

## Dynamic Analysis of Naive Adaptive Brain-Machine Interfaces

**Kevin C. Kowalski**

*nargle@nsplab.org*

**Bryan D. He**

*bryanhe@nsplab.org*

*Neural Signal Processing Laboratory, Department of Radiology,  
University of California, Los Angeles, CA 90095, U.S.A., and  
Department of Computer Science, California Institute  
of Technology, Pasadena, CA 91125, U.S.A.*

**Lakshminarayan Srinivasan**

*ls2@nsplab.org*

*Neural Signal Processing Laboratory, Department of Radiology,  
University of California, Los Angeles, CA 90095, U.S.A.*

The closed-loop operation of brain-machine interfaces (BMI) provides a context to discover foundational principles behind human-computer interaction, with emerging clinical applications to stroke, neuromuscular diseases, and trauma. In the canonical BMI, a user controls a prosthetic limb through neural signals that are recorded by electrodes and processed by a decoder into limb movements. In laboratory demonstrations with able-bodied test subjects, parameters of the decoder are commonly tuned using training data that include neural signals and corresponding overt arm movements. In the application of BMI to paralysis or amputation, arm movements are not feasible, and imagined movements create weaker, partially unrelated patterns of neural activity. BMI training must begin naive, without access to these prototypical methods for parameter initialization used in most laboratory BMI demonstrations.

Naive adaptive BMI refer to a class of methods recently introduced to address this problem. We first identify the basic elements of existing approaches based on adaptive filtering and define a decoder, ReFIT-PPF to represent these existing approaches. We then present Joint RSE, a novel approach that logically extends prior approaches. Using recently

---

K.K. and L.S. conceived of and designed the research. K.K. and B.H. performed the experiments. K.K., B.H., and L.S. analyzed the results of the experiments and prepared the figures. K.K. and L.S. drafted the manuscript. K.K., B.H., and L.S. edited and revised the manuscript and approved the final version of the manuscript. L.S. is the corresponding author.

developed human- and synthetic-subjects closed-loop BMI simulation platforms, we show that Joint RSE significantly outperforms ReFIT-PPF and nonadaptive (static) decoders. Control experiments demonstrate the critical role of jointly estimating neural parameters and user intent. In addition, we show that nonzero sensorimotor delay in the user significantly degrades ReFIT-PPF but not Joint RSE, owing to differences in the prior on intended velocity. Paradoxically, substantial differences in the nature of sensory feedback between these methods do not contribute to differences in performance between Joint RSE and ReFIT-PPF. Instead, BMI performance improvement is driven by machine learning, which outpaces rates of human learning in the human-subjects simulation platform. In this regime, nuances of error-related feedback to the human user are less relevant to rapid BMI mastery.

## 1 Introduction

---

Recent demonstrations illustrate the remarkable ability for electronics to bypass damaged neural circuits, allowing paralyzed and amputee users to control anthropomorphic robotic limbs and other assistive devices (Hochberg et al., 2012; McFarland, Sarnacki, & Wolpaw, 2010; Schalk et al., 2008). These developments represent the earliest stages of neuroscience and engineering research in brain-machine interfaces (BMI), with applications to stroke, trauma, degeneration, and other neuromuscular disease mechanisms. Significant breakthroughs in the understanding of BMI algorithm design are still needed to facilitate the elementary level of performance required for routine activities like eating, bathing, and interacting with loved ones. In this letter, we study the way in which human subjects and neural signal processing algorithms learn the basic mapping from neural signals to assistive movement, with the goal of better understanding the sensorimotor and algorithmic basis for this process.

**1.1 Definition and Categorization of Naive Adaptive Brain-Machine Interfaces.** Many BMI training paradigms involve an initial period of parameter tuning. In this period, parameters of a neural signal model are adjusted to relate observed neural signals with overt movements (Santhanam, Ryu, Yu, Afshar, & Shenoy, 2006; Serruya, Hatsopoulos, Paninski, Fellows, & Donoghue, 2002), or instructed motor imagery (Bradberry, Gentili, & Contreras-Vidal, 2011; Hochberg et al., 2006; Kim, Simeral, Hochberg, Donoghue, & Black, 2008). During this initial period, the user is not directly operating the BMI.

In contradistinction, naive adaptive control refers to algorithms that immediately engage the user in BMI operation during parameter tuning. The word naive, defined previously for the BMI literature (Gage, Ludwig, Otto, Ionides, & Kipke, 2005), indicates that BMI algorithm parameters are

randomized when subjects begin operating the BMI. The word *adaptive* indicates that these parameters are adjusted from this random initialization in an attempt to improve overall BMI performance. The BMI parameters presented in this letter are the magnitude and preferred direction in cosine-tuned motor cortical neuron point-process models (Truccolo, Eden, Fellows, Donoghue, & Brown, 2005).

Naive adaptive control is a key concept in the analysis and design of BMI because it addresses four potential barriers to clinical viability. First, actual movements are not available for BMI training in amputation or paralysis. Second, instructed motor imagery may generate patterns of neural activity that differ from patterns elicited at output neurons during closed-loop BMI control, resulting in performance degradation (Shenoy, Krauledat, Blankertz, Rao, & Muller, 2006; Taylor, Tillery, & Schwartz, 2002). Third, artificial and natural somatosensory feedback may further distort these observed neural signal patterns relative to instructed motor imagery, such as in the difference between sensorimotor potentials evoked during imagined versus overt arm movements (Miller et al., 2010) or word repetition (Leuthardt et al., 2012) that drive sensory feedback (touch, pressure, proprioception, audition, vision) from the arm, mouth, larynx, eye, and ear. Fourth, learning proficient BMI operation with nonadaptive (static) filters is slow, requiring weeks to months for basic cursor control alone (Ganguly & Carmena, 2009; Wolpaw, McFarland, Neat, & Forneris, 1991). Naive adaptive control could substantially accelerate the learning process for patients. All of these potential barriers to clinical viability are topics of ongoing research and active debate.

We divide existing naive adaptive approaches into two groups. Category 1 algorithms are based on adaptive filters (Dangi et al., 2011; Gage et al., 2005; Orsborn, Dangi, Moorman, & Carmena, 2012). The user's goals are explicitly defined by a training exercise, and these goals are related to the observed neural activity to infer neural signal parameters. Category 2 algorithms are inspired by reinforcement learning. Here, the user's goal is represented implicitly through error signals that are recorded from the brain (Gurel & Mehring, 2012; Mahmoudi & Sanchez, 2011). BMI parameters are tuned to minimize future occurrences of error signals.

Our focus is category 1 naive adaptive BMI. Existing category 1 algorithms typically use Kalman filters (Dangi et al., 2011; Gage et al., 2005; Orsborn et al., 2012).

Previously employed in a rat model of reaching movement using auditory tones (Gage et al., 2005), the category 1 approach to naive adaptive BMI was subsequently adopted in a primate model using nonnaive parameter initialization based on overt arm movements (Gilja et al., 2010, 2012) where a monkey controls an on-screen cursor with neural activity that relates to the intended cursor velocity. Related adaptive recursive Bayesian filters, not originally described for use in naive adaptive training, were previously developed for tracking neural parameters in BMI and scientific applications

(Eden, Frank, Barbieri, Solo, & Brown, 2004; Li, O'Doherty, Lebedev, & Nicolelis, 2011; Srinivasan, Eden, Mitter, & Brown, 2007).

In the category 1 method, variously named cursorGoal (Gilja et al., 2010) or ReFIT-KF (Gilja et al., 2012), the monkey is assumed to have perfect knowledge of the current position of the cursor and the on-screen target in forming its intended velocity. The ReFIT-KF algorithm rotates its estimate of intended velocity (based on the monkey's neural activity) toward the target when adjusting its parameters. This rotation explicitly assumes that intentions manifested in motor neural activity reflect zero-effective sensorimotor delay. Zero-effective delay might approximately occur as a result of predictive internal models that attempt to compensate intrinsic delays in neural systems (Golub, Yu, & Chase, 2012) or delays in the machine such as algorithms that bin neural data (Lagang & Srinivasan, 2013).

Variants of the ReFIT-KF approach were subsequently proposed by others as a solution for naive adaptive BMI in primate (Dangi et al., 2011; Orsborn et al., 2012). While ReFIT-KF adjusted parameters only intermittently (Gilja et al., 2010, 2012), these related studies (Dangi et al., 2011; Orsborn et al., 2012) examined variants of ReFIT-KF to study how the frequency of parameter updates affected learning. For example, Dangi et al. (2011) applied the ReFIT-KF approach at every time step, which we call continuous-ReFIT-KF. In performance comparisons presented in this letter, we use a point-process version of continuous-ReFIT-KF as a representative benchmark for existing category 1 naive adaptive BMI. We call this benchmark method ReFIT-PPF, where PPF indicates the use of an approximate discrete-time point-process filter (Eden et al., 2004) instead of Kalman filter (KF) variants as used by Gilja et al. (2010, 2012) and by Dangi et al. (2011) and Orsborn et al. (2012).

**1.2 Contributions of This Letter.** We now summarize the contributions of this letter, which are focused on understanding and improving category 1 naive adaptive BMI (Dangi et al., 2011; Gage et al., 2005; Orsborn et al., 2012). We first deconstruct the basic elements of ReFIT variants within a Bayesian framework (see Figure 2) to reveal three implicit design choices made in their construction. These design choices are the row labels in Figures 2 and 3. We then logically extend these design choices to create Joint RSE, a new method for naive adaptive BMI (see Figure 2). We also implement additional methods (see Figures 2 and 3) that are specifically constructed to probe the relative importance of these design choices in any category 1 naive adaptive BMI model system. The analysis demonstrates that Joint RSE outperforms ReFIT-PPF in the rate of target acquisition.

To compare these methods, we employ a model system based on healthy human volunteers, previously validated in comparison with nonhuman primate experiments (Cunningham et al., 2011). This model system translates arm movements from the subject into simulated primary motor cortical spiking activity to reproduce closed-loop behavior in moving a cursor to a

target in a two-dimensional on-screen work space (see Figure 1). We also modify this model system in two ways. First, we substantially decrease the cost of implementation by using the Microsoft Kinect 3D camera (see Figure 1B) for hand tracking (currently US\$100) instead of the Northern Digital Polaris tracking system (currently estimated at US\$60,000). Second, we demonstrate that human motor learning can be permitted by initializing neural parameters to effect a visuomotor rotation (see Figures 7–9 and 10B).

We show that Joint RSE significantly outperforms ReFIT-PPF, random walk, and static decoders (see Figures 5A and 8). Control experiments with human subjects demonstrate that Joint RSE outperforms ReFIT-PPF by jointly estimating neural parameters and user intent (see Figure 5C). In that experiment, Lockstep RSE/RSE is constructed as a lockstep version of Joint RSE to isolate the contribution of joint estimation to performance in Joint RSE. We perform further analysis using a simplified variant of our recently described stochastic control model of humans in closed-loop BMI (Lagang & Srinivasan, 2013). This analysis suggests that nonzero sensorimotor delay in the human subject significantly degrades ReFIT-PPF, while Joint RSE is robust under various levels of delay (see Figure 11). This may occur as a result of differences in the prior on intended velocity (see Figure 2).

Paradoxically, substantial differences in sensory feedback between these methods do not contribute to differences between Joint RSE and ReFIT-PPF (see Figure 5B), even under experimental conditions that permit human learning (see Figures 7–9). Further analysis reveals that the timescale of BMI performance improvement in the naive adaptive methods closely matches rates of machine learning, where human learning is undetected (see Figures 6B and 6C) or more gradual (see Figures 8B and 8C) in these experiments. The relatively slow rate of human learning helps to explain why overall BMI performance was insensitive to sensory feedback sent to the user. In this model system, machine learning far outpaced the human's ability to learn from error signals, regardless of sensory feedback.

For neurophysiologists and clinicians, this work provides testable category 1 naive adaptive decoders (see Figures 2 and 3), explaining why Joint RSE is expected to dominate ReFIT variants in the final clinical application. This letter also explicitly identifies major category 1 design choices like joint estimation, prior on intention, and sensory feedback, offering experimentally verifiable predictions on their relative importance, as well as explicit algorithm formulations to use in experimental testing.

For BMI algorithmists, this work clarifies implicit assumptions of existing category 1 naive adaptive BMI. We illustrate a new method, Joint RSE, based on the logical relaxation of these assumptions. Our analysis contributes to a growing body of work that seeks to uncover the design principles of naive adaptive BMI for the benefit of patients limited by stroke, neuromuscular diseases, and trauma.

## 2 Methods

---

**2.1 Human-Subjects Closed-Loop BMI Simulator.** The bulk of our analysis in this letter (see Figures 4–10) is based on studying able-bodied human subjects engaged in operating a closed-loop BMI simulator. This human-subjects closed-loop simulator was previously developed elsewhere with detailed comparison to primate-based BMI (Cunningham et al., 2011). In this simulator, the role of a neural control network in a target patient (see Figure 1A) that ultimately determines motor-cortical output is played by the healthy human subject in this model system (see Figure 1B). This simulator provides a viable laboratory platform for BMI design that engages actual human sensorimotor behavior (Cunningham et al., 2011). The underlying model of primary-motor-cortical activity draws on empirically derived point-process models (Moran & Schwartz, 1999; Truccolo et al., 2005) to simulate the output layer of neurons recorded by the BMI system. The human-in-the-loop aspect of this model system provides a realistic biological simulation of sensorimotor learning and online correction.

Our implementation advances this prior work in two important ways. First, we make the system affordable and accessible by using a Microsoft Kinect (currently US\$100) for markerless arm tracking instead of the Northern Digital Polaris optical tracking system (currently estimated at US\$60,000) employed previously (Cunningham et al., 2011). We have also released our MATLAB wrappers for the open-source Kinect drivers to help readers implement this simulation platform (Kowalski, 2012). Implementation is discussed in section 2.1.1.

Second, we modify the initial conditions of the human-subjects closed-loop simulator to allow human sensorimotor learning (see Figures 7–9 and 10B). Our approach is based on the visuomotor rotation task, previously employed in a study of motor learning (Krakauer & Mazzoni, 2011). In this task, visual feedback about movement is rotated by a fixed angle. For example, in a task involving point-to-point two-dimensional reaching movements with an on-screen cursor, the velocity of the cursor can be rotated by 70 degrees clockwise. In attempting reaching movements under this visual rotation, subjects can learn to adjust their arm movements to compensate this rotation based on errors they observe through the visual feedback of a computer display (Krakauer & Mazzoni, 2011). Our modification based on visuomotor rotation is discussed in section 2.1.2.

*2.1.1 How Our Kinect-Based Human-Subjects Closed-Loop Simulator Works.* In our version of the previously described human-based model system (Cunningham et al., 2011), we use recently developed open-source motion-capture code (OpenNI and PrimeSense NITE) for the Kinect 3D camera to digitize arm movements made by healthy human subjects. Although the Kinect specification describes a motion capture rate of 30 Hz, we observed that the Kinect-based software wrapper for MATLAB occasionally caused

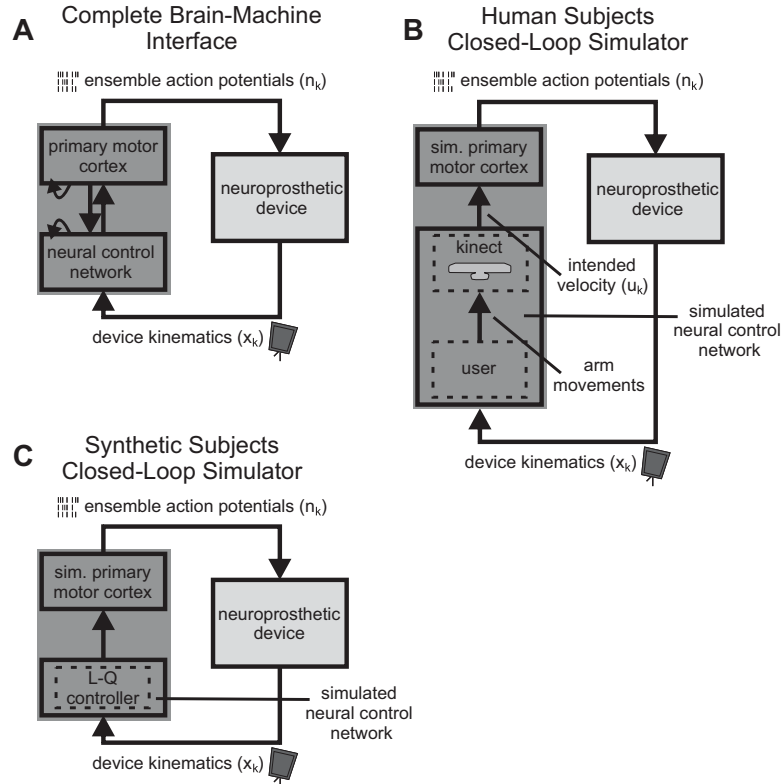


Figure 1: Closed-loop brain-machine interface (BMI) operation in practice and with two models. (A) Actual BMI system. The subject controls the BMI through an output layer with tens of primary motor cortical neurons, driven by inputs from a larger neural control network, with various recurrent connections. (B) Model system for closed-loop BMI operation based on human subjects. Here, the neural control network is represented by a healthy human subject, observing on-screen cursor kinematics, and adjusting arm movements captured by the Kinect, where arm velocity in the plane orthogonal to the camera represents intended velocity ( $u_k$ ). An empirically derived cosine-tuned point-process model of motor cortical neurons converts intended velocity into spiking events from 25 neurons. Actual and decoder estimated neural parameter values are redrawn at the beginning of every learning period. (C) Model system for closed-loop BMI operation based on a synthetic subject implemented by a linear-quadratic controller, modified from the recently described original stochastic optimal control model for closed-loop BMI operation (Lagang & Srinivasan, 2013).

the first time step of every trial to hang for 150 ms, which was generally imperceptible to the user. This event was detected and discarded in calculating arm velocities. Our code for interfacing MATLAB to the Kinect for motion



capture is freely available online, together with a brief tutorial (Kowalski, 2012).

The human subject's arm movements specify an internal representation of intended velocity. For simplicity, this velocity is captured in the 2D plane that is orthogonal to the Kinect, where  $v_x$  and  $v_y$  are velocities in orthogonal directions. The user's 2D arm velocity drives a standard empirically derived cosine-tuned point process model of motor-cortical activity (Moran & Schwartz, 1999; Truccolo et al., 2005). The conditional intensity  $\lambda(k|v_x, v_y)$  of this point-process model defines the probability with which a neuron generates a spike for the intended arm movement at time step  $k$  in terms of  $v_x$  and  $v_y$ :

$$\lambda(k|v_x, v_y) = \exp(\beta_0 + \beta_1 (v_x^2 + v_y^2)^{1/2} \cos(\theta - \theta_F)) \quad (2.1)$$

$$= \exp(av_x + bv_y + c). \quad (2.2)$$

This relationship can be expressed equivalently in polar or Cartesian form, where  $c = \beta_0$ ,  $a = \beta_1 \cos \theta_p$ , and  $b = \beta_1 \sin \theta_p$ . History dependence in spiking patterns (Truccolo et al., 2005) can be readily accommodated, as illustrated previously (Srinivasan et al., 2007).

All experiments include an ensemble size of 25 neurons, with tuning curve parameters drawn at random with every new learning session. In our purely randomized initial conditions (see Figures 4–6), parameters were chosen to result in a baseline firing rate drawn uniformly from 10 to 20 spikes per second, and a maximum firing rate drawn uniformly from 25 to 40 spikes per second at a speed of 20 cm/sec. This corresponds to  $\beta_0 \in [2.3, 3]$ ,  $\beta_1 \in [0.0112, 0.0693]$ , and  $\theta_p \in [0, 360^\circ]$ , where units on these parameters are concordant with the use of cm/sec for velocity and spikes/sec for firing rate.

Because this model does not specify a maximum firing rate, fast arm movements can drive neurons to fire at unrealistically high rates. To counteract this problem, Cunningham et al. (2011) set a maximum firing rate. Individual time bins are sized to match the maximum firing rate so that they most likely contain either 0 or 1 spikes. Consequently, spike count is reasonably simulated as a Bernoulli random variable with event probability modulated by intended velocity. Here, we choose a maximum allowed firing rate of 30 spikes per second, a reasonable approximation for primary-motor-cortical neurons (Richardson, Borghi, & Bizzi, 2012; Truccolo et al., 2005), although this is not an actual hard upper bound in the brain. This maximum rate also matches the temporal resolution of the Kinect system, which acquires arm coordinates at approximately 30 Hz.

Spike simulation and decoding was performed on a desktop computer (3.4 GHz Intel Quad Core, 16 GB RAM), with a total latency less than 30 ms, accommodating real-time performance with the 30 Hz Kinect refresh rate. Decoded cursor movements were displayed to the user on a standard LCD



monitor with 60 Hz refresh. Visual feedback was rudimentary, depicting two-dimensional cursor movements rendered in MATLAB.

*2.1.2 Simulator Modification Based on Visuomotor Rotation to Permit Human Learning.* We modified the human-subjects simulator to permit human learning by adapting the visuomotor rotation task, previously employed in the study of motor learning (Krakauer & Mazzoni, 2011). To achieve this, we changed the BMI parameter initial conditions as follows.

Recall that in Figures 4 to 6 and 10A, BMI neuron parameters are drawn at random for 25 neurons with  $\beta_0 \in [2.3, 3]$ ,  $\beta_1 \in [0.0112, 0.0693]$ , and  $\theta_p \in [0, 360^\circ]$ , as described in section 2.1.1. In our modification (see Figures 7–9 and 10B), decoder estimates of preferred direction in a subset  $R$  of the 25 neurons are rotated by a single angle from their true values rather than randomly assigned. This single angle of rotation is uniformly drawn at the beginning of each learning session from  $[-75^\circ, -60^\circ] \cup [60^\circ, 75^\circ]$ . For these neurons in  $R$ , the decoder  $\beta_0$  and  $\beta_1$  parameters are fixed at their true values. Decoder parameters for all other neurons are generated as before.

In our human-subjects closed-loop simulator experiments with partially rotated initialization described above (see Figures 7–9 and 10B), we chose  $R = 8$  of the 25 neurons (32% of neurons rotated). Based on preliminary testing, this strikes a balance between a trivial task (100% rotated) and an unreasonably hard task (0% rotated) over 26 trials within a learning session where machine learning is frozen.

**2.2 Adaptive Point-Process Filter.** This section is purely a review of an approximate adaptive point-process filter, originally described elsewhere (Eden et al., 2004), which we provide for readers' convenience. This review also introduces a consistent set of variables and filter equations that will subsequently appear in our unified perspective (see Figures 2 and 3) on various naive adaptive BMI training methods, including previously described variants (Dangi et al., 2011; Orsborn et al., 2012) of ReFIT (Gilja et al., 2010, 2012) and our proposed method, Joint RSE.

The point-process filter translates spiking neural activity into estimates of user intent that drive changes in the state of the assistive device, such as cursor velocity. It also uses this neural activity to estimate parameters of neuron tuning curves. Although experiments performed in this letter refer to spiking neural activity, the concepts introduced here apply directly to any biological signals that reflect user intent, including electroencephalography (EEG) and electromyography (EMG).

*2.2.1 State Equations and Observation Models.* The point-process filter is a type of recursive Bayesian estimation that requires a latent (hidden) variable model (also called a state equation) and an observation model. The latent variable is a random vector  $x_k$  that includes either user intention or neural parameters, or both. The state equation describes how the latent variable is expected to evolve one time step into the future. The observation model

describes the relationship between the latent variable and neural activity. In particular, the observation model specifies the probability of observing a pattern of spiking across the neural ensemble at time step  $k$ , which is determined by each neuron's tuning curve, embodied in the conditional intensity function introduced in equation 2.2.

One example of a state equation is the discrete-time linear gaussian process, indexed by time step  $k$ :

$$x_{k+1} = F_k x_k + w_k, \quad (2.3)$$

where  $F_k$  is a state evolution matrix and  $w_k$  is zero-mean gaussian noise with covariance matrix  $Q_k$ . Each training method uses a different state equation or set of state equations, which we discuss in section 2.4.

**2.2.2 Filter Equations.** Filter equations specify how observations are used to compute estimates of the latent variable. In our example, these equations determine how spiking activity results in a cursor movement *and* how neural tuning curve parameters are learned. A more expansive introduction to the approximate adaptive point-process filter equations is provided elsewhere (Eden et al., 2004). In this section, we review only the essential equations, using the random walk state equation as a basic example.

The neural activity observed at time step  $k$  is denoted  $n_k$ , and the latent variable is denoted  $x_k$ . The history of neural activity is denoted  $H_k = [n_{k-1} \ n_{k-2}, \dots, n_1]$ . Define the prediction density  $p(x_{k+1}|H_{k+1})$  with mean  $x_{k+1|k}$  and variance  $W_{k+1|k}$ . Define the posterior density  $p(x_{k+1}|n_{k+1}, H_{k+1})$  with mean  $x_{k+1|k+1}$  and variance  $W_{k+1|k+1}$ .

The filter equations are divided into a prediction step, equations 2.4 and 2.5, to calculate  $x_{k+1|k}$  and  $W_{k+1|k}$  and a subsequent update step, equations 2.6 and 2.7, to calculate  $x_{k+1|k+1}$  and  $W_{k+1|k+1}$ :

$$x_{k+1|k} = F_k x_{k|k}, \quad (2.4)$$

$$W_{k+1|k} = F_k W_{k|k} F_k^T + Q_k, \quad (2.5)$$

$$(W_{k+1|k+1})^{-1} = (W_{k+1|k})^{-1} + \sum_{j=1}^N \left[ \left( \frac{\partial \log \lambda_k^j}{\partial x_k} \right) [\lambda_k^j \Delta] \left( \frac{\partial \log \lambda_k^j}{\partial x_k} \right)^T - (n_k^j - \lambda_k^j \Delta) \frac{\partial^2 \log \lambda_k^j}{\partial x_k \partial x_k^T} \right]_{x_{k+1|k}}, \quad (2.6)$$

$$x_{k+1|k+1} = x_{k+1|k} + W_{k+1|k+1} \sum_{j=1}^N \left[ \left( \frac{\partial \log \lambda_k^j}{\partial x_k} \right) (n_k^j - \lambda_k^j \Delta) \right]_{x_{k+1|k}}. \quad (2.7)$$

In equations 2.6 and 2.7,  $\lambda_k^j$  is the conditional intensity function in equation 2.2, corresponding to neuron  $j$  evaluated at  $x_{k+1|k}$ .  $n_k^j$  is the number of spikes (0 or 1) produced by neuron  $j$  at time step  $k$ , and  $\Delta$  is the time step duration, set at 33 ms in our experiments.

**2.2.3 Joint Estimation versus Lockstep Estimation.** When neural signal parameters and user intention are both unknown, these latent variables can be estimated using equations 2.4 to 2.7. In joint estimation, parameters and intention are simultaneously estimated by including both quantities in the same latent variable vector  $x_k$ . This has the beneficial property of allowing uncertainty in neural parameters to inform estimates of user intent. For example, if neural parameters are poorly learned, this is reflected in a large covariance on the posterior density of the neural parameters and user intention is estimated with greater reliance on the state equation. Conversely, if user intention is poorly known, neural parameters are adjusted more cautiously.

Under joint estimation, the latent variable (state vector) that includes both parameters and kinematic intention in the 2D work space is

$$x_k = [a_{1k} \ b_{1k} \ c_{1k} \ a_{2k}, \dots, c_{Nk} \ p_k^x \ p_k^y \ v_k^x \ v_k^y]^T, \quad (2.8)$$

where  $a_{ik}$ ,  $b_{ik}$ , and  $c_{ik}$  refer to the three neural signal parameters (see equation 2.2) for cosine-tuned motor neuron  $i$  at time step  $k$ . The last four entries of the state vector,  $p_k^x$ ,  $p_k^y$ ,  $v_k^x$ , and  $v_k^y$ , represent the intended 2D position and velocity, respectively.

In lockstep estimation, these quantities are sequentially estimated at each time step, using equations 2.4 to 2.7 separately for estimating intent and parameters. In the first stage, user intention corresponding to latent variable  $x_k = [p_k^x \ p_k^y \ v_k^x \ v_k^y]^T$  is estimated by assuming that the current estimate of neural parameters is exact. In the second stage, neural parameter estimates corresponding to latent variable  $x_k = [a_{1k} \ b_{1k} \ c_{1k} \ a_{2k}, \dots, c_{Nk}]^T$  are updated by assuming that the current estimate of user intention is exact. These stages can be reversed. The design philosophy underlying lockstep estimation, called certainty equivalence, is the pervasive approach in BMI algorithm design, including all methods based on training data that use overt movements (Santhanam et al., 2006; Serruya et al., 2002), or instructed motor imagery (Bradberry et al., 2011; Hochberg et al., 2006; Kim et al., 2008), as well as previously developed naive adaptive control methods (Dangi et al., 2011; Gage et al., 2005; Orsborn et al., 2012). Certainty equivalence means that when parameters are estimated, the current estimate of intent is assumed to be the true intent (i.e., equivalent to being known with certainty), and vice versa (Bertsekas, 2005).

**2.3 Closed-Loop Filtering.** The language of the state equation can be adjusted slightly to reflect conditions under closed-loop BMI operation. This realization is a more recent advance (Dangi et al., 2011; Gilja et al., 2010, 2012) that followed earlier uses of the recursive Bayesian framework (Srinivasan et al., 2007, Srinivasan, Eden, Willsky, & Brown, 2005, 2006; Yu et al., 2007). At time step  $k$ , the on-screen cursor reflects  $x_{k|k}$ . Under the assumption that sensory feedback is adequate to provide the user a faithful representation of the on-screen cursor state, we can assume that the user's intention for time step  $k + 1$  is based on the cursor state in time step  $k$ . As such, the state equation, 2.3, should actually express  $x_k$ , the user's intention at time step  $k + 1$ , in terms of the actual on-screen cursor state  $x_{k|k}$ . This subtle adjustment now incorporates sensory feedback into the recursive Bayesian framework for BMI design. Although this adjustment advance (Dangi et al., 2011; Gilja et al., 2010, 2012) represents a new BMI design strategy, practically speaking, this change simply results in reducing  $W_{k|k}$  to zero in equation 2.5. In practice, we found that a small, nonzero assigned value for  $W_{k|k}$  (such as with entries on the diagonal measuring  $10^{-5} \text{ cm}^2$  or  $10^{-3} \text{ cm}^2/\text{s}^2$ ) was needed to ensure numerical stability in our MATLAB implementation.

**2.4 Naive Adaptive BMI Training Paradigms.** In this section, we discuss five active BMI paradigms that we construct to understand why naive adaptive control works and dissect the relative importance of joint estimation versus sensory feedback in this process. Each training method varies in three respects: joint estimation versus lockstep estimation, state equation used to perform neural parameter estimation, and state equation used to determine on-screen cursor position (see Figures 2 and 3). All methods update neural signal parameter estimates at the resolution of neural signal acquisition, which is 33 ms in our simulation framework.

*2.4.1 ReFIT-PPF: Representative for Existing Category 1 Naive Adaptive BMI.* The ReFIT (also called cursorGoal) training method (Gilja et al., 2010, 2012) and its variants (Dangi et al., 2011; Orsborn et al., 2012) use lockstep estimation. In this two-step process, a cursor estimation filter first determines the on-screen cursor position. Subsequently a parameter estimation filter updates neural parameter estimates. The frequency of alternation between cursor estimation and parameter estimation may determine rates of learning (Orsborn et al., 2012). For simplicity of exposition, this letter focuses on the implementation where cursor and parameter estimates are updated at each time step (Dangi et al., 2011).

The cursor estimation filter produces an estimate  $[v_{k|k}^x, v_{k|k}^y]$  for the velocity at time step  $k$  from neural activity  $n_k$  by assuming that the current estimate of the neural parameters is correct. This estimate is used to determine the on-screen cursor movement. The parameter estimation filter

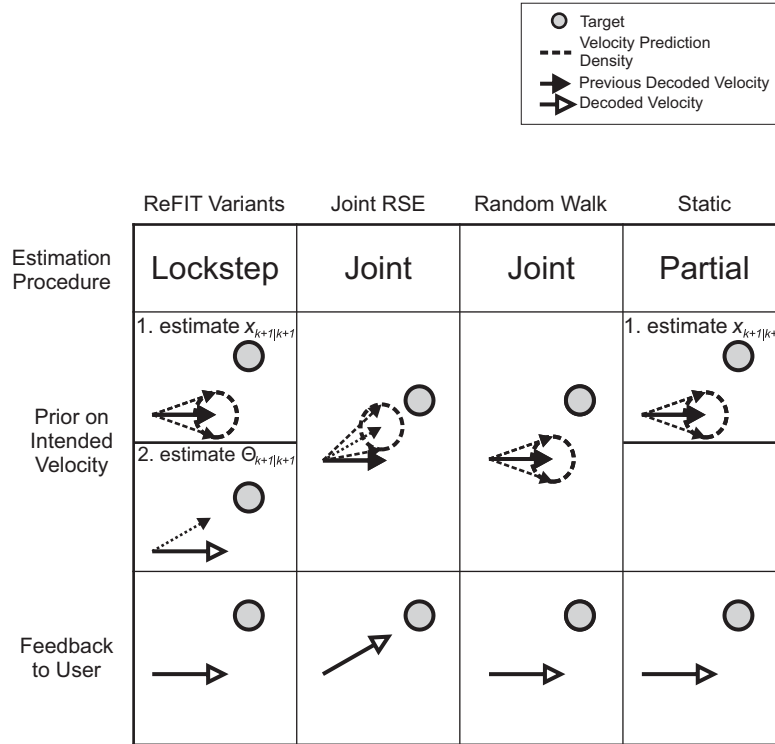


Figure 2: Naive adaptive control variants with directed and undirected priors. The ReFIT variants, Joint RSE, RW, and static training methods differ in three elemental ways, as listed in the row labels: joint versus lockstep estimation, prior on intended velocity (also called the state equation or latent variable model), and the control of visual feedback to the user (cursor movement). The RW uses an undirected prior, where ReFIT-PPF and Joint RSE use different directed priors, as defined in section 3. The various training paradigms are explained in detail in section 2.

updates the neural parameters. To accomplish this,  $[v_{k|k}^x, v_{k|k}^y]$  is rotated to point toward the target, retaining its magnitude. This estimate of intended velocity is then used in the lockstep filter component that updates neural parameters.

A rationale for using the original unrotated  $[v_{k|k}^x, v_{k|k}^y]$  to determine on-screen cursor movement is that it provides error feedback that informs the user about the BMI algorithm parameters. It is suggested that error feedback could potentially accelerate human learning. Note that running the cursor estimation filter alone without parameter updates is equivalent to static training (Ganguly & Carmena, 2010). The rationale for rotating

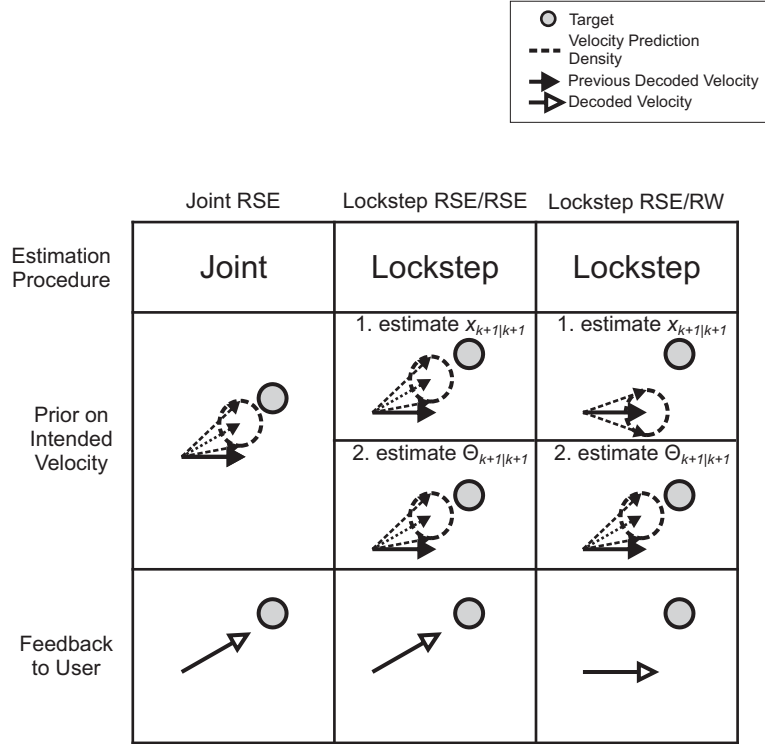


Figure 3: Naive adaptive control variants to dissect the relative importance of joint estimation versus sensory feedback. To understand the relative contribution of joint estimation and feedback to improved naive adaptive control with Joint RSE, we constructed two control methods. Lockstep RSE/RSE is nearly identical to Joint RSE except that lockstep estimation is used. Lockstep RSE/RW differs from Joint RSE in the use of lockstep estimation and the determination of cursor movement by a random walk prior (rather than the reach state equation). In its control of feedback (cursor movement), the Lockstep RSE/RW is identical to ReFIT-PPF.

$[v_{k|k}^x, v_{k|k}^y]$  in the parameter estimation filter is that neural firing reflects the user's intention to move from the current cursor position toward the known target during training, so neural parameters should be tuned to reflect this known task constraint.

Implicit in this logic is zero effective sensorimotor delay—the assumption that neural signals representing user-intended velocity instantly reflect the displacement vector between cursor position and target. Despite intrinsic brain and machine hardware delays, the user might attempt to achieve zero effective sensorimotor delay in adopting control strategies that exploit

internal models of closed-loop dynamics (Golub et al., 2012). Figure 11 in our analysis will illustrate that the zero effective sensorimotor delay assumption is a major vulnerability in ReFIT that is mitigated by our proposed method, Joint RSE.

For the cursor estimation filter, the state vector takes the form

$$x_k = [p_k^x \quad p_k^y \quad v_k^x \quad v_k^y]^T. \quad (2.9)$$

The matrices  $F_k$  and  $Q_k$  in equations 2.4 and 2.5 are

$$F_k = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.10)$$

$$Q_k = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \delta & 0 \\ 0 & 0 & 0 & \delta \end{bmatrix}, \quad (2.11)$$

for all  $k$ , with  $\Delta = 33$  ms and  $\delta = 1 \times 10^{-3}$  m<sup>2</sup>/s<sup>2</sup>. During the update step, equations 2.6 and 2.7, the neural parameter estimates  $a_{1k}b_{1k} \dots$  are assumed to be the correct values of the neural parameters when evaluating the partial derivatives of  $\lambda$ . The resulting velocity estimate  $[v_{k|k}^x, v_{k|k}^y]^T$  determines cursor movement.

For the parameter estimation filter, the state vector takes the form

$$x_k = [a_{1k} \quad b_{1k} \quad c_{1k} \quad a_{2k}, \dots, c_{Nk}]^T. \quad (2.12)$$

In this filter,  $F_k$  in equation 2.4 is the  $3N \times 3N$  identity matrix for all  $k$ , and  $Q_k$  in equation 2.5 is zero for all  $k$ . During the update step, the velocity  $[v_{k|k}^x, v_{k|k}^y]^T$  decoded by the cursor estimation filter is rotated to point in the direction of the target (while preserving decoded magnitude). This rotated velocity is used in place of the decoded velocity when evaluating  $\lambda_k^j$  and its derivatives in equations 2.6 and 2.7 for the parameter estimation filter.

**2.4.2 Joint RSE: Our Proposed Generalization of Category 1 Naive Adaptive BMI.** In this section, we propose a novel method for naive adaptive BMI that we call Joint RSE, a method that represents a combination of two prior innovations. We previously introduced the reach state equation (RSE) as a minimalistic state-space description of reaching movements (Srinivasan et al., 2006). The RSE is equivalent to a discrete-time directed random walk. Alternatively, it can be viewed as the conditional distribution on random walks given observations of the target, computed using a Riccati equation,



as seen in the Kalman filter. Elsewhere, approximate discrete-time joint estimation was developed to adaptively track neural signal parameters (Eden et al., 2004); the basic framework for this method is reviewed in section 2.2. We had previously combined joint estimation and the RSE in illustrating the capability of a novel approximate point-process filter for adapting to changing neural signal parameters (Srinivasan et al., 2007), which is different from the problem of naive BMI training. As such, the mathematical development presented in this section is essentially contained in this prior work (Eden et al., 2004; Srinivasan et al., 2006, 2007); familiarity with these papers is recommended for analyzing this letter in detail. The novel methodological insight in our work is the realization that Joint RSE represents a generalized solution of existing category 1 naive adaptive BMI. Our extensive analysis (see Figures 5–11) is directed at uncovering the basic design principles that allow Joint RSE to outperform existing category 1 naive adaptive BMI algorithms in our model systems.

Joint RSE uses joint estimation (as with the RW method) and a reach state equation (RSE) prior. The RSE provides a loosely constrained probabilistic model for how the trajectory of a reaching movement evolves over time, given partial or complete knowledge of the target location (Srinivasan et al., 2006). The RSE is the result of constraining a random walk by observations on its future state. The resulting filter simultaneously updates neural parameters and cursor kinematics using the decoded velocity to determine on-screen cursor movements. This differs from the ReFIT-PPF method in all three cardinal ways described in Figure 2: estimation procedure, prior on intended velocity, and feedback to the user.

As with the random walk method, the state vector in Joint RSE takes the form

$$x_k = [a_{1k} \quad b_{1k} \quad c_{1k} \quad a_{2k}, \dots, c_{Nk} \quad p_k^x \quad p_k^y \quad v_k^x \quad v_k^y]^T, \quad (2.13)$$

but the state equation is different. To define the state equation succinctly, we first define a number of quantities, discussed in greater detail elsewhere (Cajigas & Srinivasan, 2012; Srinivasan et al., 2006).

Let  $\tilde{F}$  be the matrix in equation 2.10, and let  $\tilde{Q}$  be the matrix in equation 2.11 so that  $\tilde{F}$  and  $\tilde{Q}$  are the state evolution and noise covariance matrices, respectively, that characterize the RW prior on cursor kinematics.

Let  $\phi(t, s)$  be the state transition matrix that advances the mean of the distribution on the state at time step  $s$  to its mean at time step  $t$ :

$$\phi(t, s) = \begin{cases} \prod_{i=1+\min(t,s)}^{\max(t,s)} \tilde{F}^{sgn(t-s)} & \text{if } t \neq s \\ I & \text{if } t = s \end{cases}. \quad (2.14)$$

Define the matrices  $\Pi(t_f, t_f)$  and  $\Pi(t, t_f)$  for any  $t < t_f$  where  $t_f$  is the total number of discrete time steps in the movement:

$$\Pi(t_f, t_f) = \Pi_f + \tilde{Q}, \quad (2.15)$$

$$\Pi(t, t_f) = \phi(t, t_f) \Pi_f [\phi(t, t_f)]^T + \sum_{i=t}^{t_f} \phi(t, t_f) \tilde{Q} [\phi(t, t_f)]^T. \quad (2.16)$$

For ease of implementation,  $\Pi(t, t_f)$  can be computed recursively, where the following recursion begins with  $t = t_f$ :

$$\Pi(t-1, t_f) = \phi(t-1, t) \Pi(t, t_f) [\phi(t-1, t)]^T + \tilde{Q}. \quad (2.17)$$

Using equation 2.17, we see that the final matrices of the Joint RSE state equation are written in terms of submatrices that correspond to neural parameter dimensions ( $F_k^{neural}$ ,  $Q_k^{neural}$ ) and cursor kinematic dimensions ( $F_k^{cursor}$ ,  $Q_k^{cursor}$ ):

$$F_k = \begin{bmatrix} F_k^{neural} & 0 \\ 0 & F_k^{cursor} \end{bmatrix}, \quad (2.18)$$

$$Q_k = \begin{bmatrix} Q_k^{neural} & 0 \\ 0 & Q_k^{cursor} \end{bmatrix}. \quad (2.19)$$

The neural parameter submatrices are the same as those in the RW:

$$F_k^{neural} = I_{3N}, \quad (2.20)$$

$$Q_k^{neural} = 0, \quad (2.21)$$

where  $I_{3N}$  is the  $3N \times 3N$  identity matrix. The cursor kinematic submatrices are

$$F_k^{cursor} = [I - \tilde{Q} \Pi^{-1}(k-1, t_{reach})] \tilde{F}, \quad (2.22)$$

$$Q_k^{cursor} = \tilde{Q} - \tilde{Q} \Pi^{-1}(k-1, t_{reach}) \tilde{Q}^T, \quad (2.23)$$

for  $k \leq t_{reach}$ , where

$$\Pi_f = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 \\ 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & \beta \end{bmatrix}, \quad (2.24)$$

with  $\alpha = 1 \times 10^{-6} \text{ m}^2$  and  $\beta = 1 \times 10^{-8} \text{ m}^2/\text{s}^2$ .

In our experiments,  $t_{reach} = 60$  time steps, which corresponds to a reach duration of  $60 \text{ time steps} \times 0.033 \text{ s/time step} = 2.0 \text{ s}$ . For  $k > t_{reach}$ , the state is assumed to evolve as a random walk, with

$$F_k^{cursor} = \tilde{F}, \quad (2.26)$$

$$Q_k^{cursor} = \tilde{Q}. \quad (2.27)$$

In successful human trials, targets were typically acquired within 1.5 seconds, so the  $k > t_{reach}$  regime was not typically entered.

As a side note, recall that the original formulation of the reach state equation implies a prediction mean  $x_{k+1|k}$  given by

$$x_{k+1|k}^{cursor} = F_k^{cursor} x_{k|k}^{cursor} + f_k, \quad (2.27)$$

where

$$f_k = \tilde{Q} \Pi^{-1}(k-1, t_{reach}) \phi(k-1, t_{reach}) x_{reach}, \quad (2.28)$$

and  $x_{reach}$  is the target kinematic vector at time  $t_{reach}$ . Because the target in our case is a static cursor position at the origin, equation 2.28 reduces to  $f_k = 0$  for all  $k$ , resulting in the linear state equation implied by equation 2.18 rather than the original affine form in equation 2.27.

**2.4.3 Random Walk: Testing the Effect of Undirected State Equations.** We define this training method as a control to examine the effect of directed versus undirected state equations in supporting parameter learning (see section 3.1). This method is a negative control and is expected to fail. The RW training method uses joint estimation, with a random walk state equation, 2.3, where this single-state equation determines both neural parameter estimation and on-screen cursor position. When we use the joint latent variable defined in equation 2.8, the matrices  $F_k$  and  $Q_k$  in this state equation are

$$F_k = \begin{bmatrix} I_{3N} & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & & \vdots & \vdots & \\ 0 & & 1 & 0 & \Delta & 0 \\ 0 & \cdots & 0 & 1 & 0 & \Delta \\ 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.29)$$

$$Q_k = \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & \delta & 0 \\ 0 & & 0 & 0 & 0 & \delta \end{bmatrix}, \quad (2.30)$$

for all  $k$ , where  $I_{3N}$  is the  $3N \times 3N$  identity matrix, with  $\Delta = 33$  ms and  $\delta = 1 \times 10^{-3} \text{ m}^2/\text{s}^2$ .

**2.4.4 Static Decoder: Testing the Capability for Pure Human Learning.** We define this training method (see Figure 2) as a control to confirm the capacity for human learning in the human subjects closed-loop simulator (see Figures 7 and 8) over the span of a single learning session. Because the static decoder involves no machine learning, any performance improvement can be attributed to pure human learning. The filter equations implement a nonadaptive point-process filter that uses randomly assigned, fixed estimates of neural signal parameters. The static decoder is identical to the RW decoder with decoder neural parameters removed from the state vector. In our example, the static decoder state vector is a  $4 \times 1$  matrix of intended position and velocity for the cursor in a two-dimensional work space.

**2.4.5 Lockstep RSE/RSE: Testing the Importance of Joint Estimation and Sensory Feedback.** We define this training method as a control to examine the contributions of joint estimation and sensory feedback in naive adaptive training. The Lockstep RSE/RSE training method is identical to the implementation of Joint RSE, except that lockstep estimation is used instead of joint estimation. Alternatively, Lockstep RSE/RSE can be viewed as identical to ReFIT-PPF, except that the reach state equation is used as the prior on intended cursor velocity, where  $F_k$  and  $Q_k$  are given by equations 2.22 and 2.23 for  $k < t_{reach}$  and equations 2.25 and 2.26 for  $k > t_{reach}$ . If Lockstep RSE/RSE performed identically to Joint RSE, we would conclude that joint estimation was noncontributory. If Lockstep RSE/RSE performed identically to Lockstep RSE/RW (defined next), we would conclude that sensory feedback was noncontributory.

**2.4.6 Lockstep RSE/RW: Testing the Role of Sensory Feedback.** This training method serves as a control to examine the effect of using sensory feedback (cursor kinematics) in naive adaptive training. This method is identical to Lockstep RSE/RSE, except that cursor movement is determined using the RW state equation. This substantially modifies the quality of feedback provided to the user, as illustrated in our online movie demonstration (Kowalski & Srinivasan, 2012). If these two lockstep methods performed identically, we would conclude that the nature of sensory feedback was

noncontributory. If we had also directly compared Lockstep RSE/RW with ReFIT-PPF in the same experimental session, we could have also directly assessed the importance of prior on intended velocity. This final comparison was not performed, mainly because of limited experimentation time, although section 3.9 indirectly addresses this point.

## 2.5 Experimental Conditions

**2.5.1 Subject Recruitment.** Ten healthy male and female volunteers, ages 18 to 22, participated in experiments with the closed-loop model system. This experimental protocol was approved by the Institutional Review Board of the University of California, Los Angeles.

**2.5.2 Task Description.** Subjects engaged in a basic reach-and-hold task. While the subject's arm movements were unconstrained in 3D, only velocity in the 2D plane orthogonal to the Kinect camera was used to drive simulated neural activity. Additionally the on-screen cursor was displayed in a 2D virtual work space. At the start of each trial, decoded cursor movements began from a random location on the starting circle. The subject would then control their arm movements to drive the cursor to a fixed circular target, centered within the starting circle. Successful trial completion required the subject to hold the cursor within the target circle for 0.5 seconds, before the trial expired at 3 seconds into the trial. Our 2D virtual work space dimensions were 20 cm for the starting circle radius, 5 cm for the target circle radius, and a cursor of negligible size. In comparison, the previously validated human-based model system (Cunningham et al., 2011) required 3D virtual movements from the origin to spherical targets of radius 2 cm, located 8 cm from the origin, using a spherical cursor of radius 2 cm.

Each subject was studied in a single session (lasting 3 to 4 hours for experiments in Figures 4 to 6 and 1 to 2 hours for experiments in Figures 7 to 9) that tested a subset of the various naive adaptive paradigms already introduced: ReFIT-PPF, Joint RSE, RW, static, Lockstep RSE/RSE, or Lockstep RSE/RW. Experiment 1 (four subjects, Figure 5A) compared RW, ReFIT-PPF, and Joint RSE. Experiment 2 (three additional, separate subjects; Figure 5B) compared Lockstep RSE/RSE and Lockstep RSE/RW. Experiment 3 (three additional, separate subjects; Figure 5C) compared Joint RSE and Lockstep RSE/RSE. Experiment 4 (two additional, separate subjects; see Figures 7 to 9 and 10B) compared all filters except RW.

Each reaching movement was performed as either a test trial or a training trial. The test trial provided an opportunity to equitably and intermittently compare performance across naive adaptive methods. During test trials, parameter learning was fixed, and all methods used the RW state equation to decode cursor movement. As a result, all methods were identical in their implementation during the test trial, except that they used different neural signal parameter values, determined by the learning process during

training trials. In the training trials, naive adaptive decoding was performed as described in the previous sections.

Each subject completed multiple learning session for each BMI training paradigm, sequenced in random fashion, while ensuring equal representation among learning paradigms. For example, in experiment 1, every set of three learning sessions contained at least one of each type of learning paradigm, although the ordering within this set was randomized. Each learning session began with a new, randomized selection of simulated motor cortical neurons and randomly configured, untrained BMI decoder parameters.

Experiments 1 to 3 (Figures 4 to 6 and 10A) involved 12 learning sessions per training paradigm per subject. Within each learning session of experiments 1 to 3, a sequence of 50 reach-and-hold trials was performed beginning with a training trial. These trials alternated in nonrandom fashion between 4 training trials and 1 test trials, for a total of 10 test trials interspersed among 40 training trials. Experiment 4 (Figures 7 to 9 and 10B) used the same organization of learning sessions, except that for brevity of experimentation, only 6 learning sessions were performed per training paradigm per subject. Also, 6 test trials were interspersed between 20 training trials, and each learning session began with a test trial. Data in Figures 8 and 9 were collected from the same two subjects but on separate days.

There was no explicit attempt to control the visual or acoustic environment beyond the presented on-screen visual feedback. In an effort to keep them alert and engaged, subjects were allowed to listen to music during the experiment. Because learning sessions alternated BMI training conditions frequently (roughly every six minutes for experiments 1 to 3 and every three minutes for experiment 4), systematic correlations between the ambient sensory environment and BMI training condition would be highly unlikely. Moreover, balanced randomization in the sequence of learning sessions would have mitigated effects of task familiarity or arousal that might artificially introduce performance differences between BMI training paradigms.

**2.6 Synthetic-Subjects Closed-Loop BMI Simulator.** In order to systematically assess the effect of sensorimotor delay on decoder performance, we used a synthetic controller based on stochastic control theory (Bertsekas, 2005) to replace the human in the loop (see Figure 1C), recalling and adapting the modeling strategy introduced in our recently published work (Lagang & Srinivasan, 2013). This controller is the standard solution to a discrete-time finite-horizon linear quadratic control problem (Bertsekas, 2005). The synthetic simulator allows us to test the case of perfectly zero effective sensorimotor delay, which is difficult to achieve with live subjects (Golub et al., 2012). The controller in the synthetic simulator (see Figure 1C) substitutes for both the human and Kinect arm tracking system (see Figure 1B). As depicted in the diagram, the controller receives the state of the

cursor as input and computes a new intended cursor velocity as output. As with the human-subjects simulator, the intended velocity in the synthetic-subjects simulator is sent to the motor-cortical neuron layer.

In our simulation, each trial lasts a maximum of 3 seconds. Time steps are simulated in 33 ms intervals. Accordingly, each trial contains a maximum of  $M = 90$  discrete time steps. In each time step  $1 \leq k \leq M$ , the synthetic controller receives a vector  $y_k$  of the current position and velocity of the cursor and outputs a  $2 \times 1$  vector of intended velocity  $u_k$ , according to the equation

$$u_k = L_k y_k. \quad (2.31)$$

In particular,  $y_k$  is the  $5 \times 1$  vector,

$$y_k = \begin{bmatrix} x_{k-1|k-1} \\ 1 \end{bmatrix}, \quad (2.32)$$

where  $x_{k-1|k-1}$  is the  $4 \times 1$  vector containing the most recent position and velocity of the cursor (or, equivalently, the decoded kinematics of the cursor from the time step  $k - 1$ ).

For each time step  $k$ , the matrix  $L_k$  is given by the equation

$$L_k = -(R_k + B_k^T K_{k+1} B_k)^{-1} B_k^T K_{k+1} A_k, \quad (2.33)$$

where  $K_k$  is given by the recursion

$$K_M = S_M, \quad (2.34)$$

$$K_k = S_k + A_k^T (K_{k+1} - K_{k+1} B_k (R_k + B_k^T K_{k+1} B_k)^{-1} B_k^T K_{k+1}) A_k, \quad (2.35)$$

and the remaining matrices are given by

$$A_k = \begin{bmatrix} 1 & 0 & \Delta & 0 & 0 \\ 0 & 1 & 0 & \Delta & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_k = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad S_k = \begin{bmatrix} \alpha_k & 0 & 0 & 0 & 0 \\ 0 & \alpha_k & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$R_k = \begin{bmatrix} \beta & 0 \\ 0 & \beta \end{bmatrix},$$

for  $1 \leq k \leq M$ . The constants were assigned as follows:  $\Delta = 0.033$  s,  $\alpha_k = 0.067$  for  $1 \leq k \leq 75$ ,  $\alpha_k = 0.33$  for  $75 < k \leq M$ , and  $\beta = 0.0083$ . Note that equations 2.33 to 2.35 are obtained directly from the classical solution to



finite-horizon linear-quadratic control problems discussed elsewhere (Bertsekas, 2005).

As a technical aside, the implementation in this letter differs from Lagang and Srinivasan (2013) in two ways, each of which reflects differences in the specific task conditions between our letter and Lagang and Srinivasan (2013). First, the task for this letter requires the reach be completed within 3 seconds to be successful, represented in the finite-horizon cost function. In contrast, Lagang and Srinivasan (2013) do not impose a completion time, reflected in the infinite-horizon cost function. Second, our task for this letter involves a naive user at the start of a learning session, represented in a control policy that assumes generic plant dynamics. In contrast, Lagang and Srinivasan (2013) simulate an approximately optimal user. There, the control policy accounts for precise dynamics resulting from the composite effect of output neural activity and the decoder in order to model closed-loop performance at the completion of training.

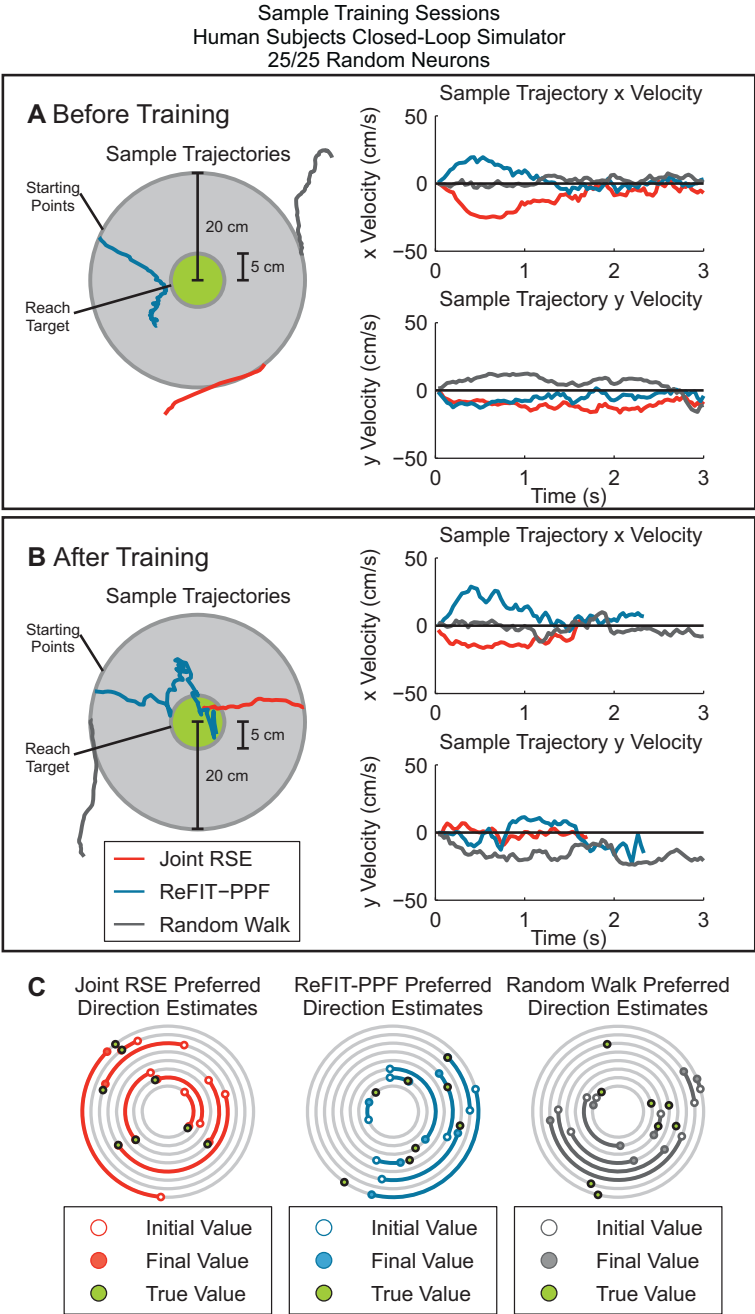
### 3 Results

---

**3.1 Category 1 Naive Adaptive Training Requires Directed Priors.** Naive adaptive control methods like ReFIT-PPF and Joint RSE learn neural signal parameters without instructed motor imagery or explicit arm movements that serve as labeled data. This capability is explained by the use of goal-directed priors. To illustrate analytically, we provide a simplified example in the appendix.

Intuitively, the directed prior serves to “probabilistically” label neural data during BMI operation. This problem formulation relates to semisupervised learning, a broad class of techniques where partially labeled data are used to infer relationships and trends (Barber, 2011). The directed prior embodies the knowledge that intention for the limb state is more likely to orient toward the target state during reaching exercises and less likely to orient elsewhere. In contrast, using the undirected prior, this probabilistic label is agnostic to the orientation of this intention, so neural signals alone cannot drive convergence of neural signal parameter estimates.

We experimentally verified whether directed priors were essential in active BMI training using the human-based model system (see section 2). Single-trial example trajectories (see Figures 4A and 4B) illustrate that naive adaptive control with undirected priors is essentially unproductive, whereas directed priors support movements that acquire the target following training. These performance differences are reflected in whether neural signal parameters representing preferred directions converge to the true preferred direction (see Figure 4C). Both ReFIT-PPF and Joint RSE decoder parameters converge appropriately, while random walk decoder parameters do not. Success rates for these various methods was examined (see Figure 5A) by aggregating across 12 learning sessions per method in each of four subjects (with each data point representing a total of 48 learning



sessions per method). This analysis confirms that performance differences exist between directed and undirected priors, where random walk success rates remain essentially flat across learning sessions. These findings were also consistent with analysis at the single-subject level (not shown).

### 3.2 Joint RSE Outperforms ReFIT-PPF in Naive Adaptive Control.

In addition, we observed that Joint RSE outperformed ReFIT-PPF in success rates, both in aggregate (see Figure 5A) and at the single-subject level (not shown). Differences in success rate emerged by the second test trial, following eight training trials. Moreover, these differences persisted to the end of the learning session, where Joint RSE success rates averaged 94%, versus 59% for ReFIT-PPF and 9% for RW. While these improvements were encouraging, we performed additional control experiments to isolate the source of these improvements.

**3.3 Major Differences in Sensory Feedback Do Not Significantly Affect Naive Adaptive Control.** In experiment 2, we examined whether substantial differences in sensory feedback between ReFIT-PPF and Joint RSE accounted for differences in performance. Accordingly, we constructed a version of Joint RSE that mimicked the ReFIT-PPF procedure (Lockstep RSE/RW) and compared it to another procedure (Lockstep RSE/RSE), which was nearly identical, except that it incorporated feedback as handled by the Joint RSE. These algorithms are described in greater detail in section 2.

Early in each training session, users experienced haphazard trajectories under the Lockstep RSE/RW procedure versus goal-directed trajectories under the Lockstep RSE/RSE procedure. In later test sessions, cursor movements in both methods would appear increasingly goal directed as a consequence of BMI training. Surprisingly, success rates were statistically indistinguishable at every test trial in the learning session (see Figure 5B). These aggregate success rates were compiled over four subjects and 12 learning sessions per method in each subject.

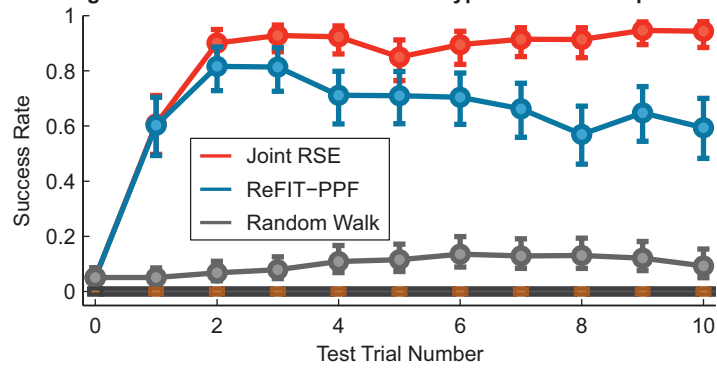
**3.4 Use of Joint Estimation Drives Performance Improvement over ReFIT-PPF.** Joint estimation represents a key procedural difference between Joint RSE and ReFIT-PPF (see Figure 2). In ReFIT-PPF, cursor movement and neural signal parameters are estimated in sequence, a process

---

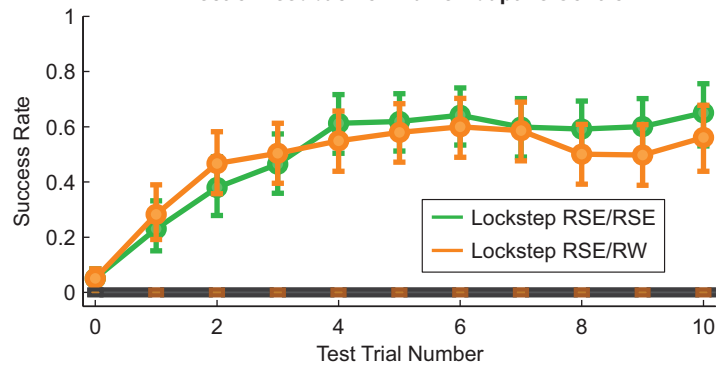
Figure 4: Single-learning-session examples of performance under naive adaptive control with directed and undirected priors. Sample trajectories and corresponding velocity profiles (A) early in the training session and (B) late in the training session. (C) Estimates of neuron preferred direction converge to true values with directed priors (ReFIT-PPF, Joint RSE), but not with undirected priors (RW) on this single learning session. Trajectories result from 25 simulated neurons and 33 ms bin width.

Human Subjects Closed-Loop Simulator  
25/25 Random Neurons

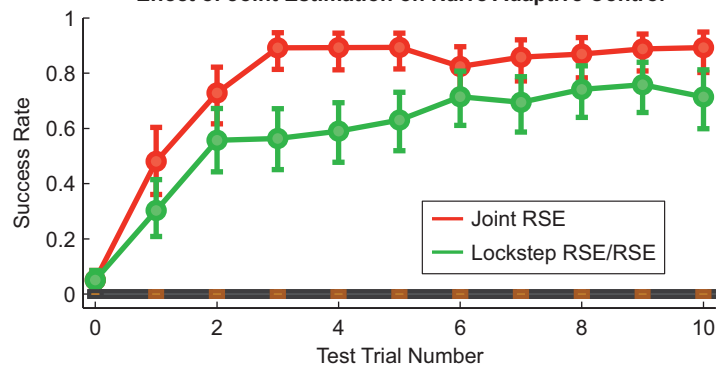
**A** Changes in Success Rate with Different Types of Naive Adaptive Control



**B** Effect of Feedback on Naive Adaptive Control



**C** Effect of Joint Estimation on Naive Adaptive Control

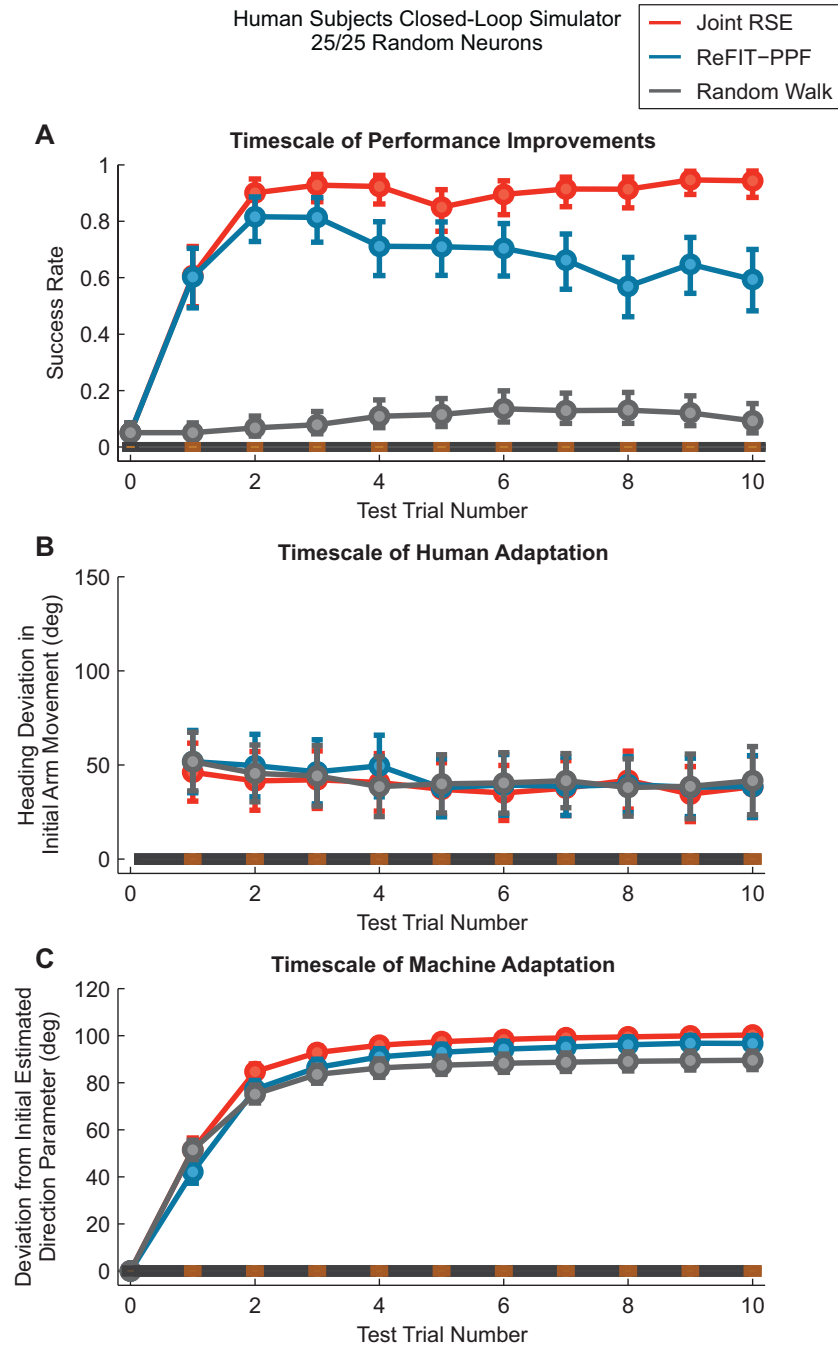


called lockstep estimation. In Joint RSE, both cursor position and neural signal parameters are estimated simultaneously, a process called joint estimation. Joint estimation allows for uncertainty in neural signal parameters to influence cursor movements, and vice versa. To determine the relevance of joint estimation to naive adaptive control, we compared versions of RSE-based naive adaptive control that were nearly identical, except that one used joint estimation (Joint RSE) and the other used lockstep estimation (Lockstep RSE/RSE). Aggregate results demonstrate significant and substantial differences in success rate over the entire learning session (see Figure 5C). In these subjects, joint estimation resulted in 90% success, while lockstep estimation resulted in 70% success using the same RSE prior.

**3.5 Naive Adaptive Control Is Dominated by Fast Timescale of Machine over Human Learning.** We next investigated whether human sensorimotor learning or machine learning predominantly accounted for rapid gains in BMI performance over these 6-minute training sessions. For each naive adaptive control method, we graphed unsigned heading deviation as a measure of human adaptation over the learning session (see Figure 6A). Unsigned heading deviation is the absolute value of the minimum

---

Figure 5: Success rate and effects of modifications on naive adaptive control. (A) Changes in success rate with naive adaptive control. Success rates and 95% confidence intervals on success rate were determined for the RW, ReFIT-PPF, and Joint RSE methods using a Bayesian procedure designed for the specific purpose of estimating learning curves (Smith et al., 2004). Four subjects participated in 12 learning sessions per method, so each data point is determined by the pooled successes and failures of 48 trials. Black and brown bars drawn near the  $x$ -axis represent alternating segments of four training trials (black) and one test trial (brown). The test trial 0 point is extrapolated from RW performance at test trial 1. (B) Effect of feedback on naive adaptive control. Success rates were not significantly different in comparison between Lockstep RSE/RSE and Lockstep RSE/RW methods, which are nearly identical methods, except in the way they apply feedback (cursor control). Lockstep RSE/RSE and Joint RSE feedback methods are identical. Lockstep RSE/RW and ReFIT-PPF feedback methods are identical. Success rates and error bars were determined as in (A). Three new subjects (different from panel A) each participated in 12 learning sessions per method, so each point is determined by the pooled successes and failures of 36 trials. Conventions are unchanged from panel A. (C) Effect of joint estimation on naive adaptive control. Success rates were significantly different in comparison between Joint RSE and Lockstep RSE/RSE methods which are nearly identical methods except in the use of joint versus lockstep estimation, respectively. Success rates and error bars were determined as in panel A. Three new subjects (different from panels A or B) each participated in 12 learning sessions per method, so each point is determined by the pooled successes and failures of 36 trials. Conventions are unchanged from panel A.



subtended angle between the subject's intended velocity and the straight-line trajectory to target. In addition, we graphed changes in BMI parameters as a function of distance to the final parameter value, a measure of machine adaptation (see Figure 6B). For convenience, we reproduced the success rate graph, which tracks how system performance changes with time (see Figure 6C). A comparison of these graphs shows that nearly 80% of performance change (see Figure 6C) and machine adaptation (see Figure 6B) is complete by the second test trial, while the human control strategy (see Figure 6A) is essentially unchanged during this time. This suggests that machine learning operates at a far more rapid timescale than human learning in this model system.

**3.6 Visuomotor Rotation Variant Permits Human Learning in Modified Human-Subjects Closed-Loop Simulator.** One possible objection to our results in sections 3.1 to 3.5 is that our experiments did not engage the capacity for human learning over the timescale of a single learning session in any method. This might limit the relevance of these results to performance in clinical applications. In response to this concern, we modified the parameter initialization procedure described in the original human subjects closed loop simulator (Cunningham et al., 2011), where BMI parameters in 32% of these neurons (8 of 25 total neurons) were initialized to a rotation from the true preferred direction. This modification is a variant on the visuomotor rotation task used elsewhere in the study of motor learning (Krakauer & Mazzoni, 2011; see section 2.1.3 for methodological details).

We employed a static filter (no machine learning; see section 2.4.4) to assay whether our subjects could exhibit learning over the timescale of a single learning session. Statistically significant human learning was exhibited in aggregate results with the static filter over two subjects (see Figure 8) as well as on a single-subject basis (not shown).

**3.7 Joint RSE Dominates ReFIT-PPF Even Under Modified Conditions When Human Learning Is Permitted.** We first plotted sample cursor

---

Figure 6: Timescales of human learning, machine learning, and BMI performance. (A) Heading deviation as a surrogate for human sensorimotor learning. Heading deviation is the minimum subtended angle between the subject's intended velocity and the straight-line trajectory to target. (B) Changes in estimated preferred direction as a surrogate for machine adaptation. Deviation from initial estimated preferred direction parameter is the minimum subtended angle between the initial estimated preferred direction and the current estimated preferred direction, averaged over all neurons. (C) Success rate, as plotted in Figure 5A, reprinted here for comparison, with timescales of (A) human and (B) machine adaptation. Subjects, trial numbers, and other conventions are unchanged from Figure 5A.



trajectories from one subject using the modified human simulator before training (test trial 0; see Figure 7A) and after training (test trial 5; see Figure 7B). For ease of visual comparison, we rotated all trajectories based on the cursor initial position so that all trajectories were presented in these graphs as movements from top to bottom. The trajectories qualitatively suggest that Joint RSE produces better control than ReFIT-PPF or static decoders. The static decoder trajectory after learning still appears qualitatively curved, where further training with the static filter might have demonstrated qualitatively straighter trajectories.

Aggregate results over two new subjects under our modified human simulator (see Figure 8) confirmed our earlier analysis that Joint RSE dominates ReFIT-PPF (see Figure 6). The differences between Joint RSE and ReFIT-PPF may have decreased, but this assertion is limited because different subjects were assayed between Figures 6 and 8. Moreover, ReFIT-PPF appears to degrade gradually over time from test trials 2 through 10 (see Figure 6), whereas our second analysis involved only five test trials (see Figure 8), which may have curtailed this gradual performance degradation.

Substantially diminished differences were seen between joint and lockstep filters (see Figure 9). Because the modified and original simulators differ only in BMI parameter initialization, these results show that parameter initialization can affect the relative importance of joint versus lockstep estimation. To understand this effect, recall that certainty equivalence means that when parameters are estimated, the current estimate of intent is assumed to be the true intent, and vice versa (Bertsekas, 2005). In our case, the modified initial conditions, chosen to allow for human learning, also inadvertently made the certainty equivalence assumption behind the lockstep filters more valid. For neurons with BMI parameters initialized to a pure rotation of their true value, the innovation term  $(n_k^j - \lambda_k^j \Delta)$  in equation 2.6 was diminished, reducing uncertainty in those neuron parameters as well as in intended velocity. This effect would not be expected in the clinical scenario because it is difficult to imagine systematically rotating estimates of neuron preferred directions without knowing their true values.

The overlap between Lockstep RSE/RSE and Lockstep RSE/RW in Figure 9 also confirmed that feedback still conferred no specific benefit under these modified conditions that permitted human learning. Where the static filter showed a steady change in the user's unsigned heading deviation (see Figure 8B), the remaining filter types exhibited no statistically detectable change in this measure (see Figures 8B and 9B).

**3.8 Other Measures of Performance.** We briefly discuss other measures of performance. The reach task explicitly and exclusively asks users to bring the cursor to the target. As such, the relevant figure of merit is target acquisition success rate, as applied in Figures 5 to 9. In Figure 10, we plot results from the various experimental conditions using mean integrated

distance to target (MID), trajectory inaccuracy, and time to target (TT). MID was introduced elsewhere (Cunningham et al., 2011).

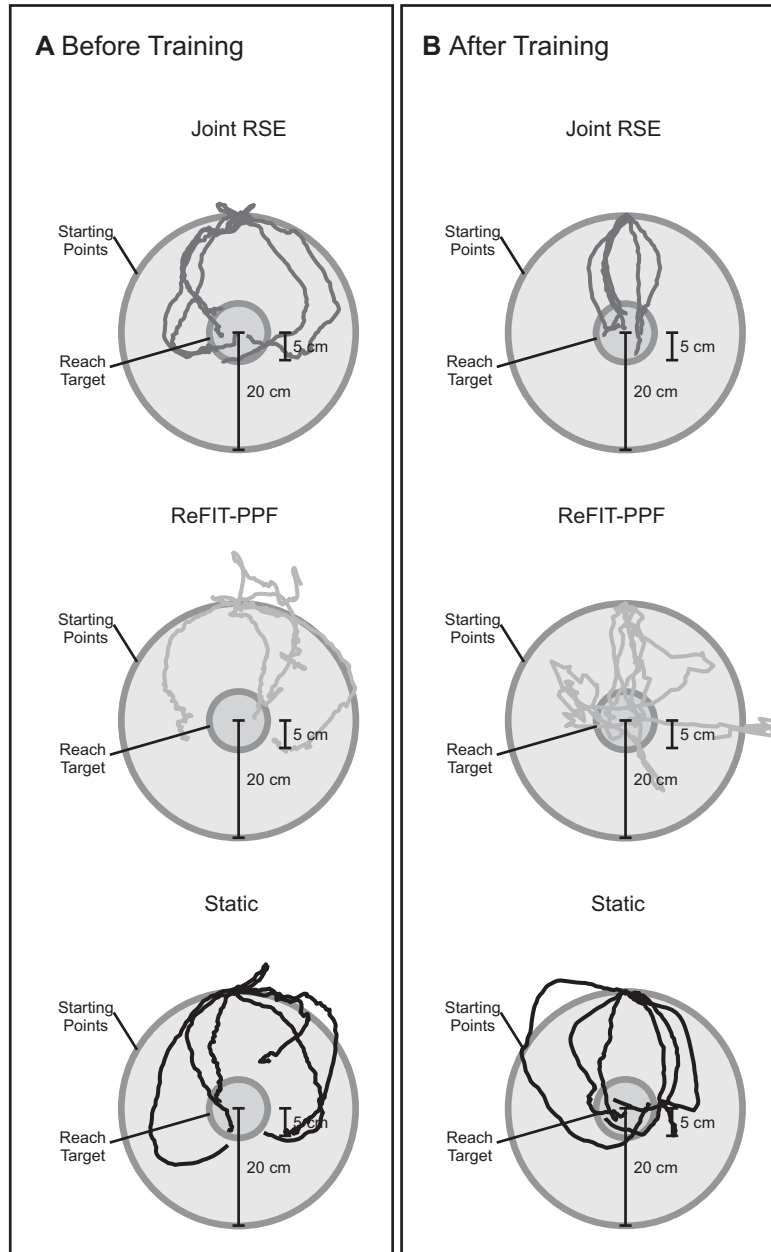
The user does not explicitly attempt to optimize any of these metrics. Ordering of performance described using a success rate is generally preserved, although trends do not reproducibly achieve significance. The notable exception is time to target. Because time to target is sensibly defined only for trials where the target is acquired (in contrast to MID and trajectory inaccuracy), the RW method appears most proficient. Although RW achieves only an approximately 10% success rate, the time to target on those trials is less than with the other methods. The case of time to target highlights the intricacy of interpreting measures that are computed on specially selected subsets of all trials.

### **3.9 Influence of Sensorimotor Delay on Differences Between Joint RSE and ReFIT-PPF Illustrated with a Synthetic Closed-Loop Simulator.**

In section 3.7, we noted that a modified human subjects simulator showed that Joint RSE dominated over ReFIT-PPF (see Figure 8) even when joint and both lockstep filters showed no differences (see Figure 9). What then could explain the residual improvement in Joint RSE over ReFIT-PPF? One possibility is the prior on intended velocity, which is distinct from the estimation procedure or choice of feedback to the user (see Figure 2). ReFIT-PPF assumes zero sensorimotor delay in its prior on intended velocity. Recall from section 2.4.1 that sensorimotor delay is the assumption by ReFIT-PPF that user-intended velocity manifests instantly in output neurons and immediately reflects the current displacement vector between cursor position and target. This would require zero delay in the brain processes that involve sensory representation and motor control. Although Joint RSE also assumes zero sensorimotor delay, it does so with uncertainty, represented in its prior on intended velocity (depicted in Figure 2) through its use of the reach state equation (Srinivasan et al., 2006).

To confirm this explanation, we would need to compare Joint RSE and ReFIT-PPF under zero sensorimotor delay. Because the human inherently has nonzero sensorimotor delay, even with the help of predictive control strategies (Golub et al., 2012), it is difficult to entirely null the sensorimotor delay in an experimental system involving human or animal subjects. Instead, we leveraged a control-theoretic model for the human operating a BMI, along the lines of our recently published work on stochastic optimal control as a theory of BMI operation (Lagang & Srinivasan, 2013). This approach is described in section 2.6. Using this purely synthetic approach to modeling BMI performance in closed loop, we systematically increased the sensorimotor delay from zero to 267 ms and 333 ms (see Figure 11). More specifically, this sensorimotor delay was introduced in the simulation purely as a sensory delay. This testing is performed with 100% randomly initialized neuron parameters rather than rotated preferred directions.

Sample Training Sessions  
Human Subjects Closed-Loop Simulator  
8/25 Random Neurons



At zero delay, ReFIT-PPF performs well because the synthetic controller points its intended velocity toward the target without delay. The zero delay assumption of ReFIT-PPF is perfectly satisfied. As sensorimotor delay increases, this assumption is increasingly violated, and ReFIT-PPF degrades rapidly. In contrast, Joint RSE does not degrade with increasing sensorimotor delay. Although the Joint RSE prior on intended velocity also points toward the target, this rotation is not asserted with certainty when estimating neural signal parameters, because the reach state equation prior inherently communicates uncertainty (see Figure 2). This representation of uncertainty accommodates the presence of sensorimotor delay and allows Joint RSE to perform well regardless of the precise value of this sensorimotor delay.

As a technical aside, note that Joint RSE and ReFIT-PPF perform equally at the zero delay condition (see Figure 11A), where the synthetic model for the human conforms perfectly to the zero sensorimotor delay assumption. Why would this be? A likely explanation is that zero sensorimotor delay permits rapid parameter learning in the first four training trials. By test trial 1, there is no uncertainty in neural signal parameters and kinematics, conforming to the certainty equivalence assumption in lockstep filters. This results in equivalence between joint and lockstep performance reflected in a convergence between Joint RSE and ReFIT-PPF in this condition. Differences in feedback between these two techniques are not a separate contributor to differences between Joint RSE and ReFIT-PPF in Figure 11 because our synthetic model for the human does not accommodate learning.

## 4 Discussion

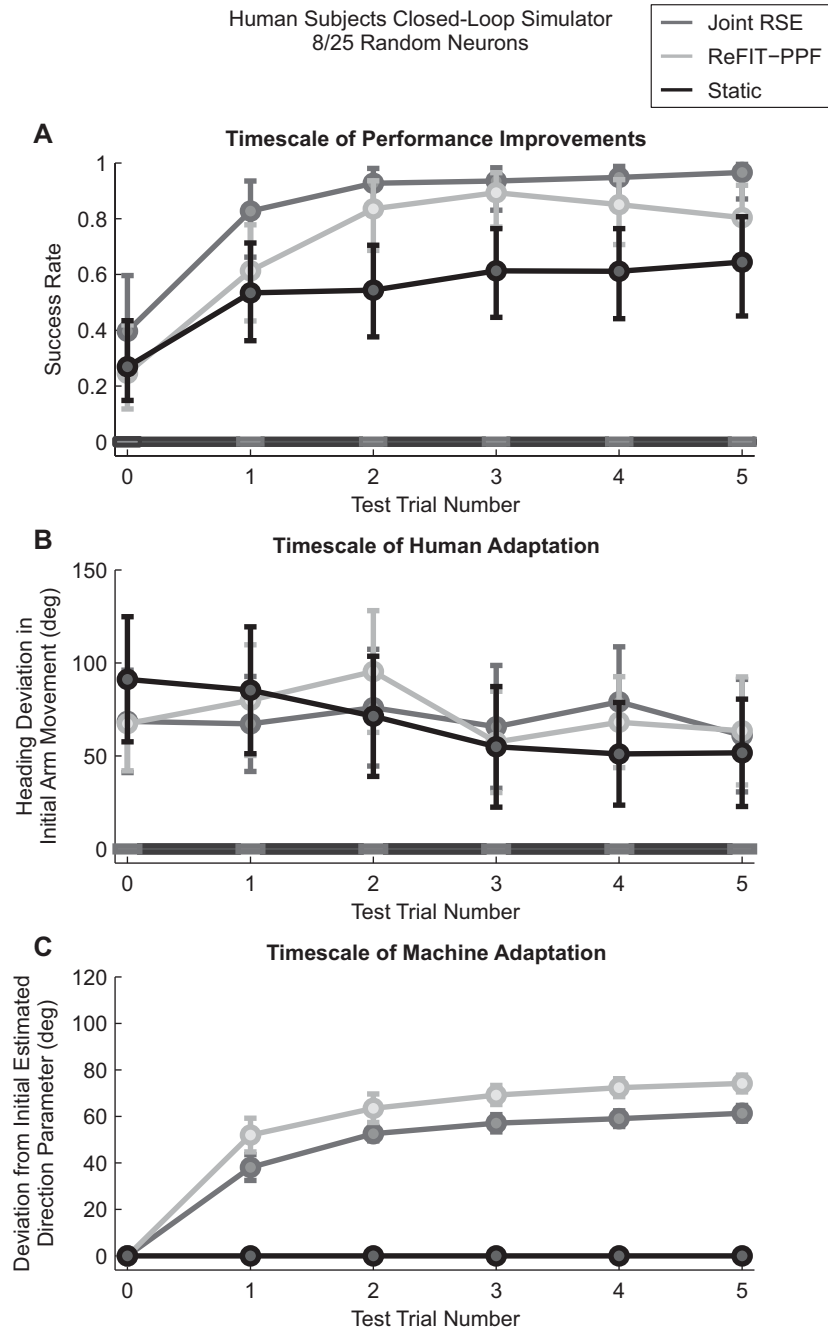
---

Section 1.2 provides a comprehensive overview of the major findings in this letter. This section proposes significant opportunities for revising category 1 naive adaptive BMI based on our findings.

**4.1 Joint Estimation Is a Key Unexploited Opportunity for Category 1 Naive Adaptive BMI.** We showed that Joint RSE primarily outperforms the ReFIT-PPF method because it jointly estimates neural signal parameters and user intentions. In contrast, prior naive adaptive methods, including ReFIT-PPF, use lockstep estimation, which sequentially updates parameters

---

Figure 7: Sample trajectories with the modified human simulator using visuomotor rotation. Position trajectories for Joint RSE, ReFIT-PPF, and static are plotted (A) before and (B) after training in one new subject. Qualitatively, Joint RSE trajectories appear smoother and more directed than ReFIT-PPF and static BMI following training. Trials begin at random positions on the outer perimeter with the target at the center. Plotted trajectories have been rotated to start at the top of the perimeter for ease of visual comparison.



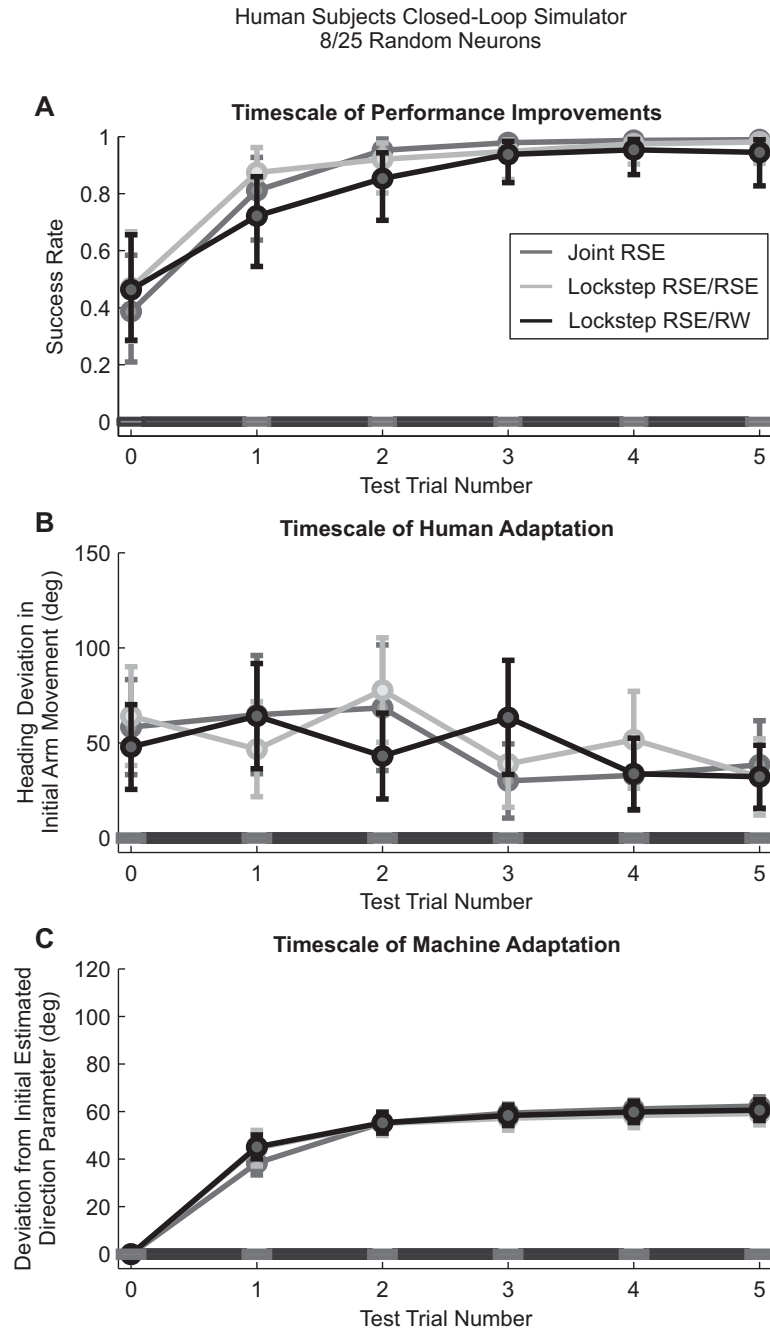
and intention (Dangi et al., 2011; Gage et al., 2005; Gilja et al., 2010; Orsborn et al., 2012). Joint estimation allows uncertainty in parameter estimates to inform the interpretation of neural signals in generating cursor movements. Early in each training session, Joint RSE relies more heavily on the RSE prior to inform cursor movements than the neural signals themselves. During this time, neural signals are used to refine parameter estimates. As uncertainty about parameter estimates decreases, Joint RSE increasingly “trusts” neural observations when steering cursor movements away from the RSE prior.

Arguably, methods that use joint estimation could result in the user feeling completely disengaged from cursor movements early in the training period when the Bayesian prior strongly determines cursor movement. This point requires further investigation. In our testing, we found Joint RSE responsive to basic user intentions even at the outset, because the baseline firing rate parameter can be estimated before any training session begins. In other words, the user could not trivially stop attending to the BMI training regimen unbeknown to the training algorithm. We preliminarily verified this (results not shown) by implementing a detector for stasis, using a mixture model based on the general purpose filter design BMI framework (Srinivasan et al., 2007). Implementing such a detector is likely to be harder with live neural recordings, where patterns of activity could be more irregular.

**4.2 Differences Between Joint RSE and Recent Naive Adaptive Methods.** How is Joint RSE different from other recently proposed naive adaptive control methods (Taylor et al., 2002; Velliste, Perel, Spalding, Whitford, & Schwartz, 2008)? Our main goal in explicitly highlighting these differences is to emphasize that performance of Joint RSE versus these other methods is not expected to be equivalent in the final clinical system. These two approaches originate from distinct conceptual foundations. The above methods represent an ad hoc (heuristic) approach, while Joint RSE is a principled (derived) approach based on Bayesian theory. In the consequent training sessions, both methods begin with decoded cursor movements that are heavily guided by the computer. Both methods also progressively decrease this guidance in transferring control to the user. However, the extent of machine versus human control in the above methods is determined through an ad hoc weighting rule. In Joint RSE, this transfer of control

---

Figure 8: Timescales of learning for Joint RSE, ReFIT-PPF, and static under the modified simulator using visuomotor rotation. These curves recapitulate the analysis in Figure 6 under the modified conditions to permit human learning over single learning sessions. Decoder neural parameters were initialized randomly for 8 of 25 neurons and to pure rotation of preferred direction in the rest. Data are aggregated from two new human subjects in the simulator, for a total of 16 learning session per technique.





occurs as a natural consequence of joint estimation, in proportion to decreasing uncertainty in estimates of the neural signal parameters. There is no single weighting parameter in the Joint RSE method.

There are innumerable other mathematical differences between the above naive adaptive methods and Joint RSE. Recent preliminary work examines a different Bayesian approach for tuning the weighting parameter in the aforementioned methods (Zhang, Schwartz, Chase, & Kass, 2012). In contrast to this new work (Zhang et al., 2012), Joint RSE does not begin from the ad hoc weighting-parameter premise of the recently developed methods. The resulting algorithms are mathematically distinct.

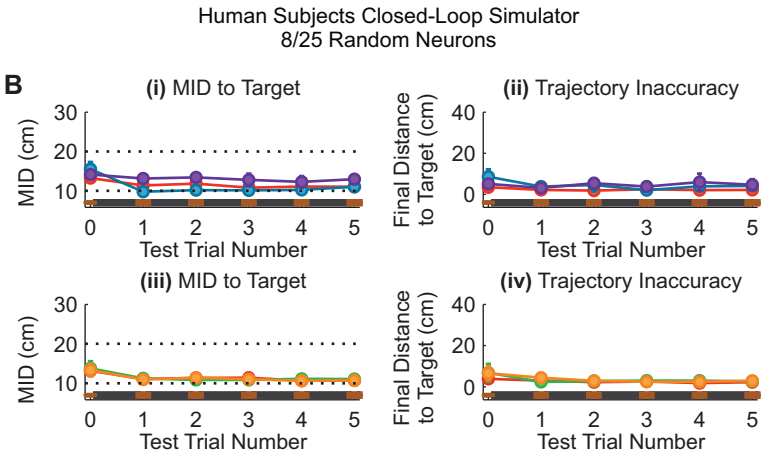
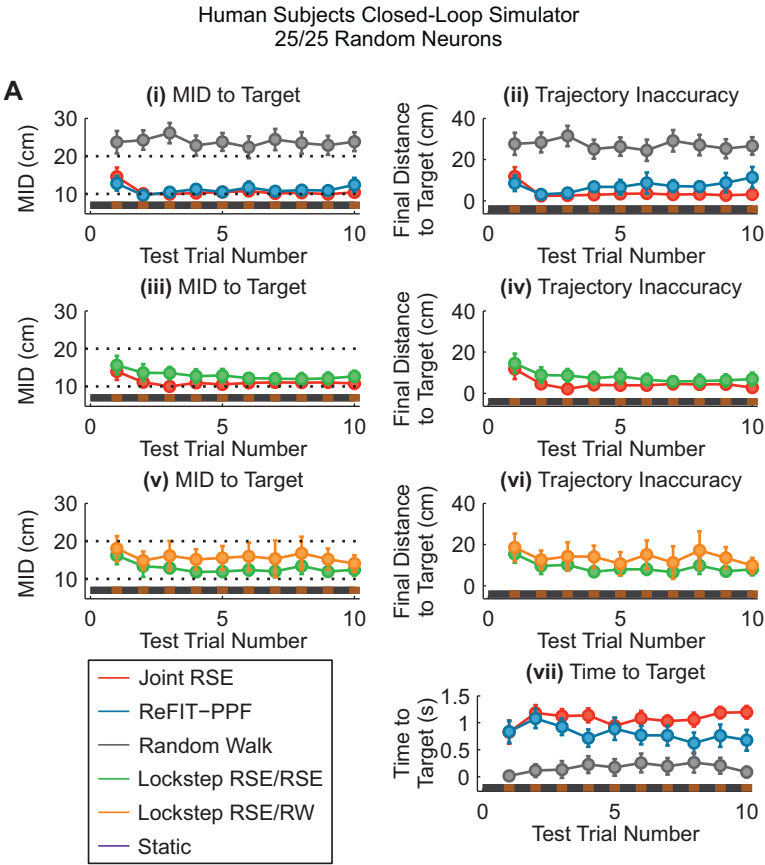
**4.3 Effective Sensorimotor Delay and BMI Algorithm Delay Are Major Design Considerations.** The delay between sensory input and response from output neurons in the patient is an intrinsic physical constraint. Preliminary work suggests that subjects appear to implement predictive control to compensate this delay but that this compensation is imperfect (Golub et al., 2012). Methods that ignore this delay could result in significant performance losses. For example, our synthetic closed-loop simulator analysis (see Figure 11) demonstrated that sensorimotor delays dramatically eroded performance in the ReFIT-PPF, a method that assumes zero sensorimotor delay. Performance with the ReFIT-PPF was decimated by a delay of 330 ms. In contrast, Joint RSE was entirely immune to this effect over this range of delays, because the prior on intended velocity accommodated an uncertain sensorimotor delay.

A second source of delay not addressed in this letter results from the neural signal algorithm itself. In existing systems, the delay can be on the order of tens to hundreds of milliseconds. A related concept is the effect of spike binning on closed-loop performance, illustrated elsewhere (Cunningham et al., 2011), and subsequently explained in our previous work using a control-theoretic model for the human in closed-loop BMI operation (Lagang & Srinivasan, 2013).

**4.4 Motivation and Reward During Training Are Important Design Considerations.** Although this letter focuses on quantifiable performance differences between Joint RSE and ReFIT-PPF, user experience is likely to be equally important in clinical practice. Our figures show that substantial

---

Figure 9: Timescales of learning for joint versus lockstep RSE methods under the modified simulator using visuomotor rotation. These curves recapitulate the analysis in Figure 8 under conditions that permit human learning within a single learning session. As with Figure 8, decoder neural parameters were initialized randomly for 8 of 25 neurons and to pure rotation of preferred direction in the rest. Data are from the same two human subjects used in Figure 8, for a total of 16 learning session per technique.



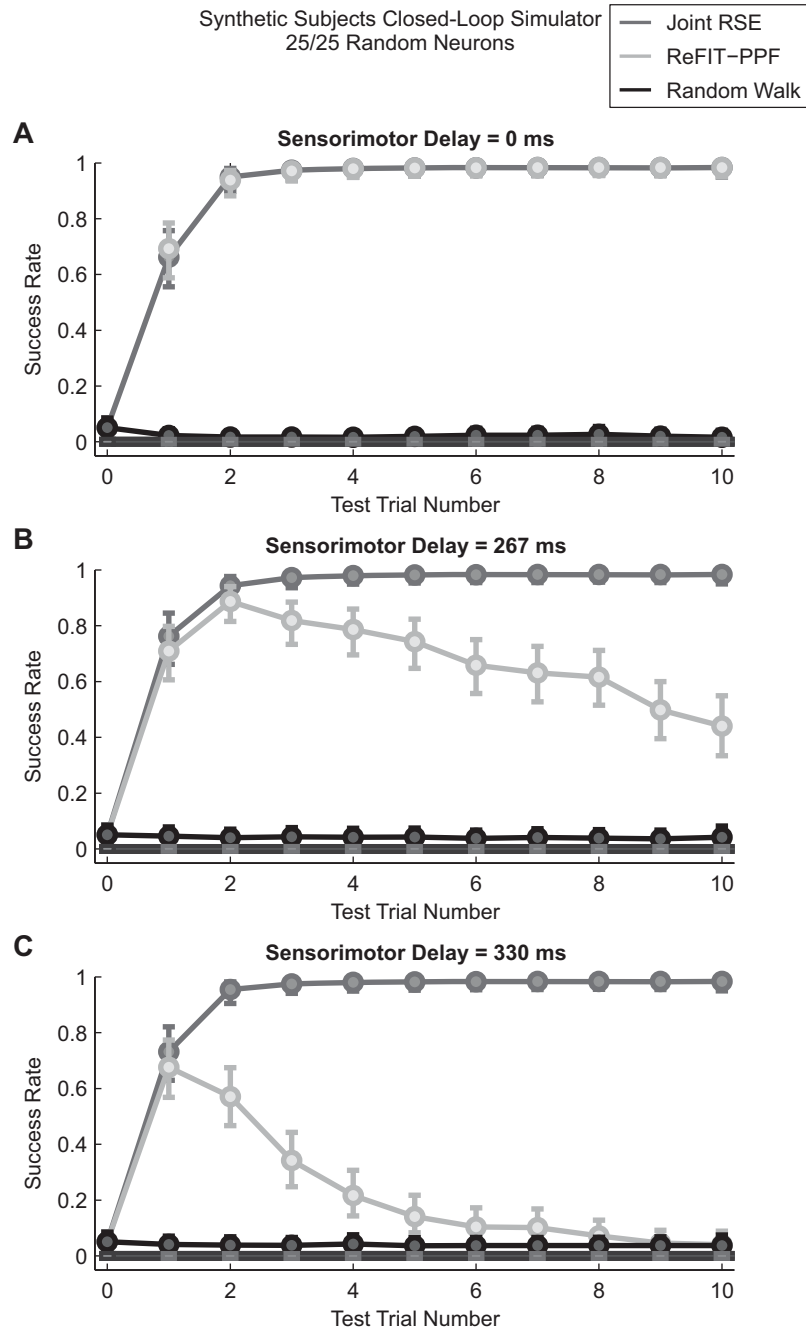
differences in visual feedback did not cause differences in overall performance. However, many of our subjects retrospectively described training sessions with ReFIT-PPF as frustrating because cursor movements were necessarily haphazard during training, resulting in many failed training trials. In contrast, training with Joint RSE was a pleasanter user experience because training trials involved smooth cursor trajectories that most often succeeded. These aesthetics are best illustrated in our online video demonstration of ReFIT-PPF versus Joint RSE (Kowalski & Srinivasan, 2012). For both experimentalists using animal models and clinicians working with patients over several hours or days, algorithms that cause high rates of failure early in training could destroy user motivation and consequently undermine the human and machine learning processes required for BMI mastery.

**4.5 Details of Sensory Feedback May Be Irrelevant to BMI Learning at Short Timescales.** Online feedback to the user regarding task performance is believed to be vital to the learning process. Adaptive controller models of BMI skill acquisition (DiGiovanna, Mahmoudi, Fortes, Principe, & Sanchez, 2009; Heliot, Ganguly, Jimenez, & Carmena, 2010) suggest that feedback may be useful even in naive adaptive control. Our results illustrate that efforts to optimize the precise choice of feedback could be irrelevant during periods of training where the timescale of machine learning far outpaces that of human learning. Conversely, when machine learning has flattened, choice of feedback could be vital to driving subsequent performance improvements that rely on human learning. Connections to the robotic stroke rehabilitation literature could be illuminating in this regard, including cautionary insights on counterproductive rehabilitation strategies (Marchal-Crespo & Reinkensmeyer, 2009).

**4.6 Masking Errors from the User During Training Could Better Coordinate Human and Machine Learning.** While both ReFIT-PPF and Joint RSE represent adaptive methods, it has been recently suggested that machine adaptation may actually disrupt the training process in some implementations (Judy, 2011; Orsborn, Dangi, Moorman, & Carmena, 2011). The intuition behind this argument is that continually changing properties of an adaptive BMI could be difficult for the user to learn. While closed-loop simulation of coexistent human and machine learning suggests that BMI training can successfully converge (DiGiovanna et al., 2009; Heliot et al.,

---

Figure 10: Other metrics of performance. (A) Mean integrated distance to target (i, iii, v), trajectory inaccuracy (ii, iv, vi), and time to target (vii) as defined in the text, using four subjects and conditions from Figure 5. (B) These measures using subjects and conditions from Figures 8 and 9.



2010), initial experimental results show that training improves by alternating between static and adaptive BMI sessions (Orsborn et al., 2011).

The fact that Joint RSE masks training errors during adaptive BMI sessions could be a favorable trait during adaptive BMI sessions. Because training errors are not available to drive human learning, the adaptive BMI session under Joint RSE represents a more purely machine adaptation block than adaptive BMI under ReFIT-PPF. Future work could use this property of Joint RSE to explore the possibility that coadaptive learning might achieve minimum training time by alternating between sessions that halt human learning and sessions that halt machine learning. For readers familiar with control theory, this notion will be reminiscent of bang-bang control, which is provably optimal in similar minimum-time control problems (Stengel, 1994).

Toward this concept of optimal control in training regimens, our analysis (see Figure 6) also illustrates the use of surrogate quantities for tracking human and machine learning rates in the experimental setting. These surrogate quantities are observable in practice for use in developing a principled rule (control policy) to switch between static and adaptive BMI training sessions or to continuously tune rates of machine learning in order to ensure that a coadaptive training regimen is productive rather than disruptive.

**4.7 Is the Out-to-Center Task a Trivial Version of the Classical Center-Out Task?** One possible concern regarding the out-to-center reaching task (see Figures 4 and 7) is that algorithms and performance analysis presented here might not generalize to arbitrary starting and ending positions. To the contrary, the out-to-center reaching task is equivalent to the center-to-out reaching task in the distribution of cursor velocities needed to achieve successful trajectories. This is because these two task types are essentially equivalent except for a change in frame of reference. When the “center” of the screen is redefined as the target location, a center-to-out reaching task becomes an out-to-center reaching task. For this reason, the various coadaptive BMI algorithms presented here extend readily to reach training paradigms with arbitrary starting and ending positions. Our out-to-center reaching task is also more difficult than the classical center-to-out reaching

---

Figure 11: Effect of sensorimotor delay assessed with synthetic subject closed-loop simulator. In contrast to prior analyses, this analysis uses a linear quadratic controller in place of the human subjects, adapted from prior theoretical work (Lagang & Srinivasan, 2013). Performances for the Joint RSE, ReFIT-PPF, and RW are compared under (A) zero delay, (B) 267 ms delay, and (C) 330 ms somatosensory delay. Specifically, output neural activity reflects on-screen cursor state from time into the past equal to the specified delay. Sensorimotor delay is the counterpart to delay studied elsewhere (Golub et al., 2012) introduced by the BMI algorithm itself.

task that involves only eight discrete, circumferentially placed target locations. Because the out-to-center reaching task initializes the cursor position to a random location on the starting circle, this is equivalent to an infinite number of possible target locations in the classical center-to-out reaching task.

Another possible concern regarding the out-to-center reaching task is that a trivial decoder with knowledge of the target location might entirely disregard neural activity and guide the cursor toward the center of the work space to achieve perfect performance. Such a decoder might not generalize to other tasks with multiple possible target locations. However, arbitrary reaching tasks also have equivalent trivial decoders. When the starting position and the final position are known, neural activity is not needed to perfectly drive the cursor from start to finish. The existence of trivial decoders is not specific to the out-to-center task. In our training sessions, we specifically avoided trivial decoders. In other words, all algorithms investigated in this study were uninformed about target location during testing periods.

**4.8 Closed-Loop Simulation Widens the Development Pipeline for Naive Adaptive BMI Design.** This letter demonstrates the application of two recently introduced closed-loop models (Cunningham et al., 2011; Lagang & Srinivasan, 2013) based on a simulated neural activity for BMI analysis and design. We first used a human-subjects closed-loop simulator, adapted from previous work that established the validity of this approach in comparison with nonhuman primate models (Cunningham et al., 2011). We demonstrated an implementation based on the Microsoft Kinect (currently US\$100), which could widen access to this model. We also showed how to modify initial conditions within this model to elicit human learning based on visuomotor rotation (Krakauer & Mazzoni, 2011).

The computer-based simulation component of our analysis (see Figure 11) adapted our recent work on modeling the patient as a stochastic controller in the closed-loop operation of BMI (Lagang & Srinivasan, 2013). This approach was essential to probe zero effective sensorimotor delay, which is difficult to study in live subjects due to intrinsic neural delays, even despite predictive control strategies (Golub et al., 2012). As such, this work adds to a small but growing body of literature that demonstrates the utility of patients who are entirely simulated by computer to understand and improve the dynamics of closed-loop BMI operation (Dangi et al., 2011; Gurel & Mehring, 2012; Heliot et al., 2010; Mahmoudi & Sanchez, 2011).

A core challenge of technology development in BMI research is that all model systems, including simulation platforms, animal models, and epilepsy patients with subdural grids, ultimately lack the complexity of disease pathogenesis in some major subset of patients that demonstrate paralysis, which ultimately limits conclusions about asymptotic performance or learning dynamics in comparisons of various algorithms.

Modeling paralysis in any of these systems (simulated, animal, or human) is especially challenging because the various pathways to paralysis are so heterogeneous, including various types of stroke, various mechanisms of trauma, and multiple pathways of neurodegeneration. As a specific example, spinal cord injury alone is heterogeneous in neurologic manifestation, affecting sensory and motor function to varying degrees depending on mechanism and anatomic extent. While the BMI literature commonly assumes complete loss of motor and proprioceptive function, anterior cord syndrome is a classically described manifestation of motor paralysis that retains proprioception by sparing the dorsal column. Common mechanisms for anterior cord syndrome include trauma, myelitis, and anterior spinal artery infarct (Blumenfeld, 2002). Ultimately model systems (simulated, animal, or human) provide a starting point for technology development that disregard these nuances of clinical presentation, where more expensive but essential testing in target patients through randomized controlled trials will be the ultimate gold standard.

**4.9 Limitations of the Study.** In carrying forward these insights, we acknowledge the full spectrum of fundamental limitations in using our human-based model system to predict behavior in the final clinical system and target patient population. First, our model requires the user to control natural arm movements, tracked by the Kinect system. In contrast, both invasive and noninvasive BMIs require the user to control a subset of neural signals measured by a specific recording modality. Because the model system and clinical system engage different user outputs, mechanisms and constraints of learning in the neural substrate may differ; this is an open question.

Second, our measure for human learning, unsigned heading deviation, is not a comprehensive characterization of the human. Some motor behaviors that could qualify as learning may not be captured by this measure. A more comprehensive approach would involve modeling the human as a control policy. For example, our recent theoretical work on stochastic control as a model for BMI (Lagang & Srinivasan, 2013) could be extended by identifying parameters of the control policy executed by the human using experimental data. This policy represents a mapping from sensory feedback to neural signals. Parameter convergence with this modeling approach is nontrivial in this letter because of limited data: every learning session involves different parameter initial conditions and consequently requires a different human control policy. A similar approach to empirical modeling of the user was briefly suggested in related work, where longer periods of nonstationarity might facilitate identification of the control policy (Golub et al., 2012).

Third, our model system involves visual feedback, whereas the target clinical application could potentially allow richer sensory feedback. Possible examples include intact native somatosensory feedback from the

paralyzed limb or artificial feedback through vibrotactile displays. This richer feedback could potentially accelerate human learning to a timescale that is sufficiently fast as to be relevant to dynamic adjustments in machine learning.

Fourth, the neural substrates of our healthy young volunteers are different from those of target patients due to innumerable disease-related effects, including cortical reorganization following trauma, cerebrovascular regulation following stroke, and metabolic changes associated with neurodegenerative diseases, which could affect the capacity for sensorimotor learning and control. There are likely other limitations of our experimental system, as all model systems are imperfect by construction, and testing in the final clinical setting with a defined target patient population remains the ultimate gold standard in BMI design.

#### Appendix: Basic Example of Necessity for Directed Priors in Category 1 Naive Adaptive BMI

---

In this appendix, we provide a simplified example that illustrates why directed priors are needed in category 1 naive adaptive BMI. Consider neural observations  $n_k$  at time step  $k$  that are related to intended 1D arm state  $x_k$  by a neural signal parameter  $\alpha$  in a simplified relationship as follows:

$$n_k = \alpha x_k \quad (\text{A.1})$$

With  $w_k \sim N(0, \sigma^2)$ , an example of an undirected prior on intended arm states is the random walk on  $x_k$ :

$$x_{k+1} = x_k + w_k. \quad (\text{A.2})$$

An example of a directed prior is a random walk on  $x_k$  with known drift  $b$ :

$$x_{k+1} = x_k + b + w_k. \quad (\text{A.3})$$

Define increments of the observation process  $z_{k+1} = n_{k+1} - n_k$ . These  $z_k$  are independent and identically distributed gaussian random variables. For the undirected prior,

$$z_k \sim N(0, \alpha^2 \sigma^2). \quad (\text{A.4})$$

For the directed prior,

$$z_k \sim N(\alpha b, \alpha^2 \sigma^2). \quad (\text{A.5})$$



In the context of the training regimen, a goal is defined to the user, effectively constraining  $b$ . For the directed prior, estimating the neural signal parameter  $\alpha$  involves computing the mean of samples  $z_k$  and dividing by  $b$ :

$$\hat{\alpha}_{ML} = \frac{\sum_{k=1}^N z_k}{b}. \quad (\text{A.6})$$

For the undirected prior,  $\alpha$  is essentially undetermined in sign, where the sample variance on  $z_k$  is dependent on  $\alpha^2$  and the sample mean is 0.

### Acknowledgments

---

K.K. was supported by funding from the UCLA Amgen Scholars Program. L.S. was supported by funding from the American Heart Association Scientist Development grant (11SDG7550015), the DARPA Reliable Neural-Interface Technology (RE-NET) Program, and the UCLA Radiology Exploratory Development Grant. We thank Alexander Wein for his help in organizing experiments and Theodore Koenig, Luis Armendariz, and Siamak Yousefi for their help in beta testing the MATLAB-Kinect interface installation procedure described in our online tutorial. We declare no competing financial interests.

### References

---

- Barber, D. (2011). *Bayesian reasoning and machine learning*. Cambridge, New York: Cambridge University Press.
- Bertsekas, D. P. (2005). *Dynamic programming and optimal control* (3rd ed.). Belmont, MA: Athena Scientific.
- Blumenfeld, H. (2002). *Neuroanatomy through clinical cases*. Sunderland, MA: Sinauer.
- Bradberry, T. J., Gentili, R. J., & Contreras-Vidal, J. L. (2011). Fast attainment of computer cursor control with noninvasively acquired brain signals. *J. Neural Eng.*, 8(3), 036010.
- Cajigas, I., & Srinivasan, L. (2012). Erratum: "A State-Space Analysis for Reconstruction of Goal-Directed Movements Using Neural Signals." *Neural Computation*, 24(4), 1106–1107.
- Cunningham, J. P., Nuyujukian, P., Gilja, V., Chestek, C. A., Ryu, S. I., & Shenoy, K. V. (2011). A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. *J. Neurophysiol.*, 105(4), 1932–1949.
- Dangi, S., Gowda, S., Heliot, R., Carmenta, J. M. (2011). Adaptive Kalman filtering for closed-loop brain-machine interface systems. In *Proceedings of the 5th IEEE EMBS Conference on Neural Engineering* (pp. 609–612). Piscataway, NJ: IEEE.
- DiGiovanna, J., Mahmoudi, B., Fortes, J., Principe, J. C., & Sanchez, J. C. (2009). Coadaptive brain-machine interface via reinforcement learning. *IEEE Trans. Biomed. Eng.*, 56(1), 54–64.

- Eden, U. T., Frank, L. M., Barbieri, R., Solo, V., & Brown, E. N. (2004). Dynamic analysis of neural encoding by point process adaptive filtering. *Neural Computation*, 16, 971–998.
- Gage, G. J., Ludwig, K. A., Otto, K. J., Ionides, E. L., & Kipke, D. R. (2005). Naive coadaptive cortical control. *J. Neural Eng.*, 2(2), 52–63.
- Ganguly, K., & Carmena, J. M. (2009). Emergence of a stable cortical map for neuroprosthetic control. *PLoS Biol*, 7(7), e1000153.
- Ganguly, K., & Carmena, J. M. (2010). Neural correlates of skill acquisition with a cortical brain-machine interface. *J. Mot. Behav.*, 42(6), 355–360. doi:10.1080/00222895.2010.526457
- Gilja, V., Nuyujukian, P., Chestek, C. A., Cunningham, J. P., Yu, B. M., Fan, J. M., et al. (2012). A high-performance neural prosthesis enabled by control algorithm design. *Nat. Neurosci.*, 15(12), 1752–1757. doi:10.1038/nn.3265
- Gilja, V., Nuyujukian, P., Chestek, C. A., Cunningham, J. P., Yu, B. M., Ryu, S. I., et al. (2010). A high-performance continuous cortically-controlled prosthesis enabled by feedback control design. *Sfn Abstract Viewer/Itinerary Planner*, Program/Poster 20.26.
- Golub, M. D., Yu, B. M., & Chase, S. M. (2012, Aug. 28 2012–Sept. 1 2012). *Internal models engaged by brain-computer interface control*. Paper presented at the Engineering in Medicine and Biology Society, 2012 Annual International Conference of the IEEE.
- Gurel, T., & Mehring, C. (2012). Unsupervised adaptation of brain-machine interface decoders. *Front. Neurosci.*, 6, 164. doi:10.3389/fnins.2012.00164
- Heliot, R., Ganguly, K., Jimenez, J., & Carmena, J. M. (2010). Learning in closed-loop brain-machine interfaces: Modeling and experimental validation. *IEEE Trans. Syst. Man Cybern. B Cybern.*, 40(5), 1387–1397.
- Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., et al. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398), 372–375. doi:10.1038/nature11076
- Hochberg, L. R., Serruya, M. D., Friehs, G. M., Mukand, J. A., Saleh, M., Caplan, A. H., et al. (2006). Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099), 164–171.
- Judy, J. (2011). *Reliable Central-nervous-system interfaces (RCI): A BAA in the program on Reliable Neural Technology (RE-NET)*. Defense Advanced Research Projects Agency. [https://safe.sysplan.com/renef/files/rci/2\\_Judy.pdf](https://safe.sysplan.com/renef/files/rci/2_Judy.pdf)
- Kim, S. P., Simeral, J. D., Hochberg, L. R., Donoghue, J. P., & Black, M. J. (2008). Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *J. Neural Eng.*, 5(4), 455–476. doi:10.1088/1741-2560/5/4/010
- Kowalski, K. (2012). MATLAB-Kinect interface code.
- Kowalski, K., & Srinivasan, L. (2012). *Simulation of two methods in co-adaptive control for brain-machine interfaces*. <http://hdl.handle.net/1721.1/70975>
- Krakauer, J. W., & Mazzoni, P. (2011). Human sensorimotor learning: adaptation, skill, and beyond. *Curr. Opin. Neurobiol.*, 21(4), 636–644. doi:10.1016/j.conb.2011.06.012
- Lagang, M., & Srinivasan, L. (2013). Stochastic optimal control as a theory of brain-machine interface operation. *Neural Comput.*, 25(2), 374–417. doi:10.1162/NECO\_a\_00394

- Leuthardt, E. C., Pei, X. M., Breshears, J., Gaona, C., Sharma, M., Freudenberg, Z., et al. (2012). Temporal evolution of gamma activity in human cortex during an overt and covert word repetition task. *Front. Hum. Neurosci.*, 6, 99. doi:10.3389/fnhum.2012.00099
- Li, Z., O'Doherty, J. E., Lebedev, M. A., & Nicolelis, M. A. (2011). Adaptive decoding for brain-machine interfaces through Bayesian parameter updates. *Neural Comput.*, 23(12), 3162–3204. doi:10.1162/NECO\_a\_00207
- Mahmoudi, B., & Sanchez, J. C. (2011). A symbiotic brain-machine interface through value-based decision making. *PLoS One*, 6(3), e14760. doi:10.1371/journal.pone.0014760
- Marchal-Crespo, L., & Reinkensmeyer, D. J. (2009). Review of control strategies for robotic movement training after neurologic injury. *J. Neuroeng. Rehabil.*, 6, 20. doi:10.1186/1743-0003-6-20
- McFarland, D. J., Sarnacki, W. A., & Wolpaw, J. R. (2010). Electroencephalographic (EEG) control of three-dimensional movement. *J. Neural Eng.*, 7(3), 036007. doi:10.1088/1741-2560/7/3/036007
- Miller, K. J., Schalk, G., Fetz, E. E., den Nijs, M., Ojemann, J. G., & Rao, R. P. (2010). Cortical activity during motor execution, motor imagery, and imagery-based online feedback. *Proc. Natl. Acad. Sci. USA*, 107(9), 4430–4435. doi:10.1073/pnas.0913697107
- Moran, D. W., & Schwartz, A. B. (1999). Motor cortical representation of speed and direction during reaching. *J. Neurophysiol.*, 82(5), 2676–2692.
- Orsborn, A. L., Dangi, S., Moorman, H. G., & Carmena, J. M. (2011). Exploring timescales of closed-loop decoder adaptation in brain-machine interfaces. In *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, 2011 (pp. 5436–5439). Piscataway, NJ: IEEE. doi:10.1109/IEMBS.2011.6091387
- Orsborn, A. L., Dangi, S., Moorman, H. G., & Carmena, J. M. (2012). Closed-loop decoder adaptation on intermediate time-scales facilitates rapid BMI performance improvements independent of decoder initialization conditions. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20, 468–477.
- Richardson, A. G., Borghi, T., & Bizzi, E. (2012). Activity of the same motor cortex neurons during repeated experience with perturbed movement dynamics. *J. Neurophysiol.*, 107, 3144–3154. doi:10.1152/jn.00477.2011
- Santhanam, G., Ryu, S. I., Yu, B. M., Afshar, A., & Shenoy, K. V. (2006). A high-performance brain-computer interface. *Nature*, 442(7099), 195–198.
- Schalk, G., Miller, K. J., Anderson, N. R., Wilson, J. A., Smyth, M. D., Ojemann, J. G., et al. (2008). Two-dimensional movement control using electrocorticographic signals in humans. *J. Neural Eng.*, 5(1), 75–84. doi:10.1088/1741-2560/5/1/008
- Serruya, M. D., Hatsopoulos, N. G., Paninski, L., Fellows, M. R., & Donoghue, J. P. (2002). Instant neural control of a movement signal. *Nature*, 416(6877), 141–142.
- Shenoy, P., Krauledat, M., Blankertz, B., Rao, R. P., & Muller, K. R. (2006). Towards adaptive classification for BCI. *J. Neural Eng.*, 3(1), R13–R23. doi:10.1088/1741-2560/3/1/R02
- Smith, A. C., Frank, L. M., Wirth, S., Yanike, M., Hu, D., Kubota, Y., et al. (2004). Dynamic analysis of learning in behavioral experiments. *J. Neurosci.*, 24(2), 447–461. doi:10.1523/JNEUROSCI.2908-03.2004
- Srinivasan, L., Eden, U. T., Mitter, S. K., & Brown, E. N. (2007). General purpose filter design for neural prosthetic devices. *J. Neurophysiol.*, 98, 2456–2475.

- Srinivasan, L., Eden, U. T., Willsky, A. S., & Brown, E. N. (2005). Goal-directed state equation for tracking reaching movements using neural signals. In *Proc. 2nd Internat IEEE EMBS Conf. on Neural Engineering* (pp. 352–355). Piscataway, NJ: IEEE.
- Srinivasan, L., Eden, U. T., Willsky, A. S., & Brown, E. N. (2006). A state-space analysis for reconstruction of goal-directed movements using neural signals. *Neural Computation*, 18, 465–494.
- Stengel, R. F. (1994). *Optimal control and estimation*. New York: Dover.
- Taylor, D. M., Tillery, S. I., & Schwartz, A. B. (2002). Direct cortical control of 3D neuroprosthetic devices. *Science*, 296(5574), 1829–1832.
- Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., & Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *J. Neurophysiol.*, 93(2), 1074–1089. doi:10.1152/jn.00697.2004
- Velliste, M., Perel, S., Spalding, M. C., Whitford, A. S., & Schwartz, A. B. (2008). Cortical control of a prosthetic arm for self-feeding. *Nature*, 453(7198), 1098–1101. doi:10.1038/nature06996
- Wolpaw, J. R., McFarland, D. J., Neat, G. W., & Forneris, C. A. (1991). An EEG-based brain-computer interface for cursor control. *Electroencephalogr. Clin. Neurophysiol.*, 78(3), 252–259.
- Yu, B. M., Kemere, C., Santhanam, G., Afshar, A., Ryu, S. I., Meng, T. H., et al. (2007). Mixture of trajectory models for neural decoding of goal-directed movements. *J. Neurophysiol.*, 97, 3763–3780.
- Zhang, Y., Schwartz, A. B., Chase, S. M., & Kass, R. E. (2012). Bayesian learning in assisted brain-computer interface tasks. In *Proceedings of the 34th Annual International Conference of the IEEE EMBS* (p. 2740). Piscataway, NJ: IEEE.